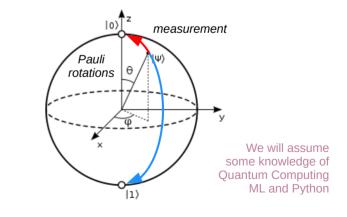


QML workshop QML team QML and aims Parameterised circuits Data encoding Angle encoding The good, the bad and the ugly State measurement Quantum model training Parameters optimisation Model geometry and gradients QML readings PennyLane demo Summary

## **Quantum Machine Learning**

Introduction

Jacob L. Cybulski Enquanted, Melbourne, Australia



Creative Commons CC BY-NC-ND





Sebastian Zając

Assistant Professor SGH Warsaw School of Economics LinkedIn



### **Tomek Rybotycki**

QML Researcher SRI PAS, NCAC PAS, CEAI AGH, KPLabs LinkedIn

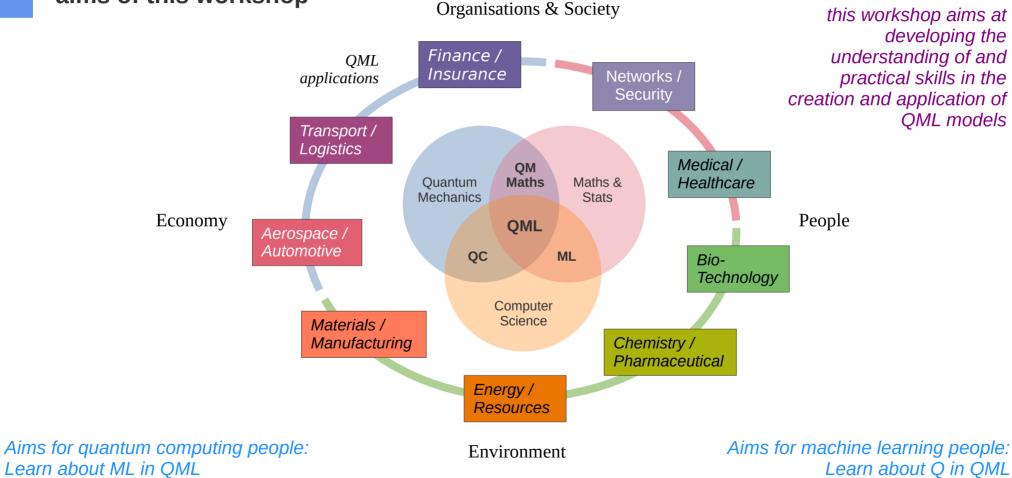
Jacob Cybulski Founder, Researcher, Consultant at Enquanted

> and Honorary A/Prof In Quantum Computing Deakin University LinkedIn



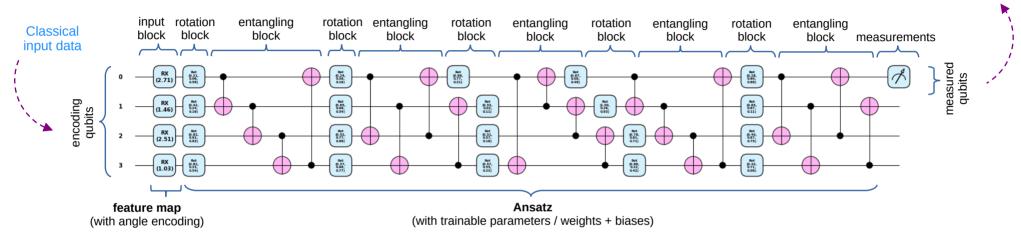
Coordinator of QPoland *and* CEO of Fundacja Quantum AI LinkedIn Jacob L. Cybulski, Quantum Business Series (Deakin, RMIT, ACS, Warsaw School of Economics) Jacob L. Cybulski, Quantum Computing Intro Series (SheQuantum, Assoc of Polish Profs in Australia) 2021-2025

## Quantum ML aims of this workshop



## Variational Quantum Models = Parameterised Quantum Circuits

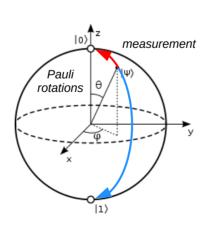
Ansatz parameters are trainable. Each parameter defines a dimension in the model's parameter space.



We can create a "variational" model = a circuit template with parameterised gates, e.g. P(a), Ry(a) or Rz(a), each allowing rotation of a qubit state in x, y or z axis (as per Bloch sphere).

Typically, but now always, the circuit consists of three blocks:

- a feature map (input)
- an ansatz (processing)
- measurements (output)



Classical input data is encoded into the feature map's parameters, setting the model's initial quantum state.

Classical output data

The quantum state is then altered by an ansatz, which consists of parameterised gates (operations), which alter the initial quantum state.

The quantum state of the circuit is then measured and interpreted as the model's output in classical data form, e.g. as binary values, integer or real value, a single event's probability or the probability distribution.

# Data encoding strategies

There are many methods of data embedding, such as: the *basis*, *angle*, *amplitude*, *QRAM*, ... encoding,

In this workshop we will rely on *angle encoding* realised as qubit state rotation by the angle defined by the data.

The rotation operators are always available in a quantum platform API (e.g. *Rx*, *Ry*, *Rz* or *Rxyz*).

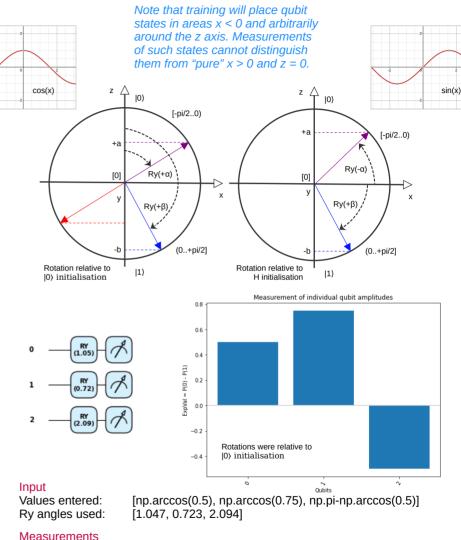
Typically, the encoding rotation is performed around x or y axis, or both (allowing two values per qubit).

Rotations are *relative to a specific qubit state*, commonly starting at  $|0\rangle$  state, or  $(|0\rangle+|1\rangle)/sqrt(2)$ , which require qubits to be initialised in these states.

The encoded value could be represented either by the *angular rotation*, or the *amplitude* of the qubit projective measurement (Z).

In some cases, input data is repeatedly encoded and interspersed with ansatz layers, called *data reuploading*, which improves the model performance.

Maria Schuld and Francesco Petruccione. Machine Learning with Quantum Computers. 2nd ed. Springer, 2021.



Probabilities: Amplitudes:

[[0.25, 0.75], [0.562, 0.438], [0.25, 0.75]] [0.5, 0.75, -0.5]

## Angle encoding The Good, the Bad and the Ugly

 $\label{eq:angle} Angle \ Range: -6 \ to \ 6 \\ 0..\pi = "blue" (long) | >\pi = "deeppink" (medium) | -\pi..0 = "darkorange" (short)$ 

This example shows encoding values wrapping around the Bloch sphere (possibly several times), so that different values are represented by the same amplitude.

(violates principle 1)

Two principles of quantum data encoding:

- distinct data values should map onto distinct amplitudes (and angular codes)
- the same data values should always map into identical amplitudes (and angular codes)

This visualisation shortened some of the vector to highlight their difference, however, they are all of length 1. Angle Range: 0 to 3.141592653589793 0.. $\pi$  = "blue" (long) | > $\pi$  = "deeppink" (medium) | - $\pi$ ..0 = "darkorange" (short)

> This example shows encoding values wrapping around the range of 0..2pi of the Bloch sphere, ensuring that different values are represented by unique amplitudes.

> > The problem may arise if we have the same value represented by two distinct amplitudes

(violates principle 2)

The red dot represents the same value which on two occasions maps onto two different angles and thus two amplitudes.

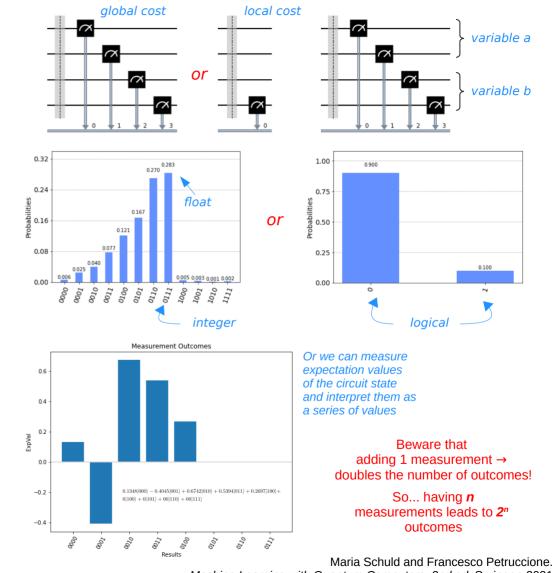
## **Commonly used** measurements and interpretation

There are many ways of obtaining the outcome of a circuit execution, e.g. we can measure:

- all qubits (global cost / measurement)
- a few selected qubits (local cost / measurement)
- groups of qubits (each as a variable value)
- as counts of outcomes (repeated measurements)
- as probabilities of outcomes (e.g. P(|0111)))
- as Pauli expectation values (i.e. of eigenvalues)
- as expectation of interpreted values (e.g. 0 to 15)
- as variance, etc.

Repeated circuit measurement can be interpreted as outcomes of different types, e.g.

- as a probability distribution (as is)
- as a series of values (via expvals)
- as a binary outcome: single qubit measurement or parity of kets
- as an integer: most probable ket in multi-qubit measurement
- as a continuous variable: probability of the selected ket (e.g. |0<sup>n</sup>>)

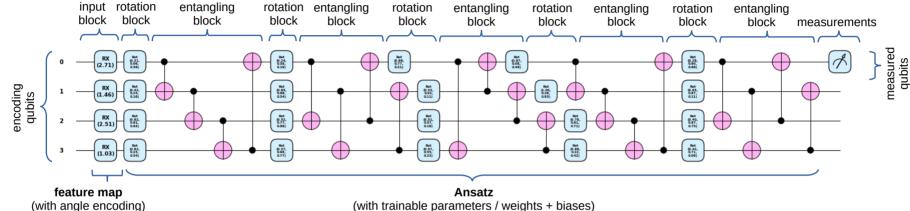


Machine Learning with Quantum Computers. 2nd ed. Springer, 2021.

## Ansatz design and training A simple quantum classifier ...

### Beware that adding qubits adds parameters and entanglements!

The number of states represented by the circuit grows exponentially with the number of qubits!



### feature maps vary in:

structure and function

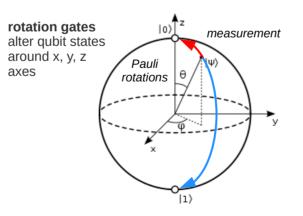
### ansatze vary in:

- width (qubits #)
- depth (layers #)
- dimensions (param #)
- structure (e.g. funnelling)
- entangling (circular, linear, sca)

### ansatz layers consist of:

rotation blocks and entangling blocks of R(x, y, z) and CNOT gates

otation) (entanglement)



To execute a circuit we just apply it to input data and the optimum parameters

### different cost functions:

R2, MAE, MSE, Huber, Poisson, cross-entropy, hinge-embedding, Kullback-Leibner divergence

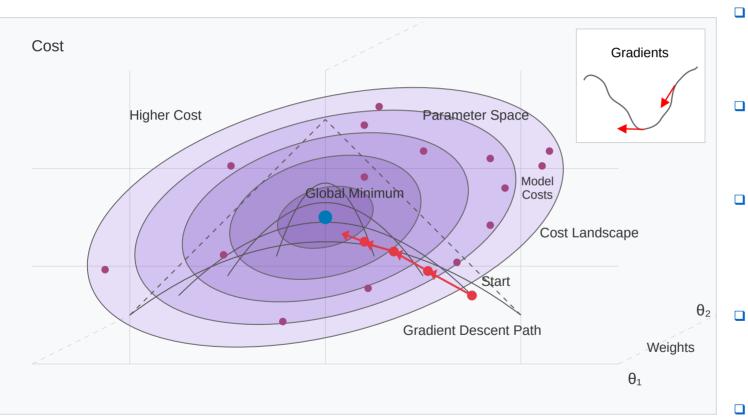
### different optimisers:

gradient based (Adam, NAdam and SPSA) linear approximation methods (COBYLA) non-linear approximation methods (BFGS) quantum natural gradient optimiser (QNG)

### circuit execution on:

simulators (CPUs), accelerators (GPUs) and real quantum machines (QPUs)

## Problem-solving with DL models Classical model optimisation

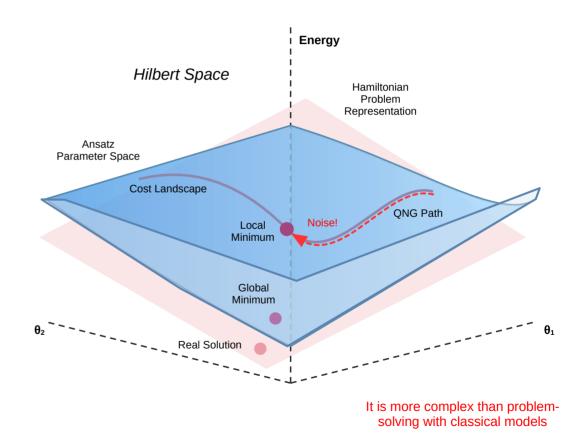


Gradients are local, i.e. their changes influence only their immediate neighbourhood

- A Deep Learning model aims to represent the problem.
- It is parameterised with weights and biases.
  - The model quality is linked to the *cost function*, where the lower the cost, the better is the model.
  - The costs of all possible model parameterisations, form a multi-dim surface, or the cost landscape.
  - The optimisation process relies on the shape of the landscape, which in turn is reflected in the gradient of points on the cost surface.
  - Gradient descent algorithms can assist in the identification of the model with the minimum cost.
- Backpropagation can also be used to very efficiently recalculate NN weights.

- The abstract mathematical model represents the problem to be solved, e.g. in the form of a *Hamiltonian*
- The Hamiltonian defines some geometry in *Hilbert space* with the optimum solution associated with the optimum energy
- The ansatz of a quantum circuit approximates the Hamiltonian and therefore the problem
- The parameters of the ansatz define a parameter space that overlaps and intersects the Hamiltonian problem space
- Our search for the problem solution is hence restricted to the ansatz parameter space
- Ideally, the selected optimiser and the cost function should understand the principles and processes of quantum models, e.g. Quantum Natural Gradient (QNG) optimiser
- The QNG method defines gradients, which are then calculated for the cost landscape (or the manifold), which spans the ansatz geometry
- The QNG optimiser can then identify a local optimum solution for this ansatz
- Noise can prevent finding the local optimum

## Problem-solving with QML Quantum model optimisation

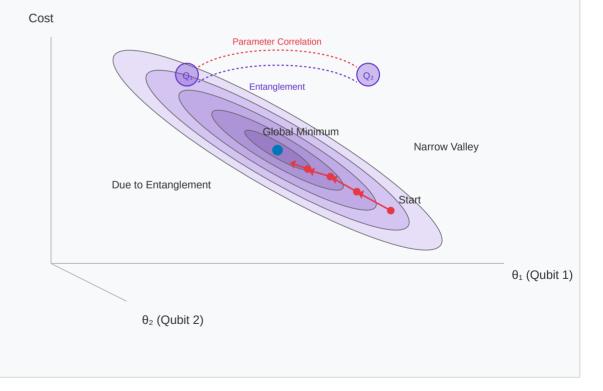


- Optimisation of quantum model needs unique approaches due to the emergence of non-local gradients
- Entangled qubits result in correlated parameters and gradients, so the changes to one are reflected in the distant others
- The cost landscape of highly entangled circuits commonly features *narrow valleys*
- Also, backpropagation cannot be used directly in training quantum circuits, as their state is not directly accessible and the measurement collapses the state
- Gradient descend can still be used with global gradients, i.e. those derived from the geometry of the cost landscape
- Stochastic optimisation techniques are highly effective when the cost landscape is smooth (no quantum moise)
- Other techniques are also available, such as *particle swarm optimisers*, these however are applicable to smaller models

## **Quantum model optimisation**

### Parameter Space with Entangled Qubit Gradients

Correlated gradient landscape due to quantum entanglement



## PennyLane Demo Everything is a function!



### PennyLane (PL) ...

- Supports differentiable programming paradigm
- Integrates seamlessly with the *Python*
- Has a range of operations for *state preparation*, *gates* and *measurements*
- Supports creation of flexible *quantum algorithms*
- Executes on *simulators* and *quantum hardware*
- Supports error mitigation
- Extends its *quantum gradients* with those from JAX, PyTorch, Keras, TensorFlow, or NumPy
- Supports hybrid quantum-classical models
- Allows training with *hardware-compatible gradients* and *higher-order derivatives*
- Provides numerous quantum models, such as: *QNNs*, *quantum kernels* and *Fourier models*
- Can be extended with models and optimisers from other SDKs, e.g. *PyTorch* and *TensorFlow*

### PennyLane Demo:

- Create a simple PL model to fit a simple function
- Learn to initialise model weights
- Explore the impact of ansatz structure on performance
- Create minimalistic quantum models
- Learn the interaction of data encoding and ansatz
- Investigate different types of entangling
- Apply the best solution to more complex data
- Learn about stamina and wisdom in QML development

### Key takeaways:

- Plan model development, tests and experiments
- Bad data encoding spoils the bunch!
- Strong entanglement improves the data fit
- More width and depth = the curse of dimensionality
- Carefully consider your quantum model initialisation
- Surprise a single qubit model still works! (and well)
- More training does not solve the problems
- Data reuploading makes a huge difference!

## **Recommended reading** on QML



## Dancing with Qubits

From aubits to algorithms, embark on the quantum computing journey shaping our future

#### Second Edition



**Robert S. Sutor** 

### Maria Schuld Francesco Petruccione

uantum Science and Technology

### **Machine** Learning with Quantum Computers

Second Edition

Springer

#### PennyLane: Automatic differentiation of hybrid quantumclassical computations

Ville Bergholm,<sup>1</sup> Josh Izaac,<sup>1</sup> Maria Schuld,<sup>1</sup> Christian Gogolin,<sup>1</sup> M. Sohaib Alam,<sup>2</sup> Shahnawaz Ahmed,<sup>3</sup> Juan Miguel Arrazola,<sup>1</sup> Carsten Blank,<sup>4</sup> Alain Delgado,<sup>1</sup> Soran Jahangiri,<sup>1</sup> Keri McKiernan,<sup>2</sup> Johannes Jakob Meyer,<sup>5</sup> Zevue Niu,<sup>1</sup> Antal Száva,<sup>1</sup> and Nathan Killoran<sup>1</sup>

1 Xanadu, 777 Bay Street Toronto, Canada

2

Modern applications of machine learning in quantum sciences

Anna Dawid<sup>1,2\*</sup>, Julian Arnold<sup>3†</sup>, Borja Requena<sup>2†</sup>, Alexander Gresch<sup>4†</sup>, Marcin Płodzień<sup>2</sup>,

Kaelan Donatella<sup>5</sup>, Kim A. Nicoli<sup>6,7</sup>, Paolo Stornati<sup>2</sup>, Rouven Koch<sup>8</sup>, Miriam Büttner<sup>9</sup> Robert Okuła<sup>10,11</sup>, Gorka Muñoz-Gil<sup>12</sup>, Rodrigo A, Vargas-Hernández<sup>13,14</sup>, Alba

Cervera-Lierta15, Juan Carrasquilla14, Vedran Dunjko16, Marylou Gabrié17, Patrick

Huembeli<sup>18,19</sup>, Evert van Nieuwenburg<sup>16,20</sup>, Filippo Vicentini<sup>18</sup>, Lei Wang<sup>21,22</sup>, Sebastian J.

Wetzel<sup>23</sup>, Giuseppe Carleo<sup>18</sup>, Eliška Greplová<sup>24</sup>, Roman Krems<sup>25</sup>, Florian Marguardt<sup>26,27</sup>,

Michał Tomza<sup>1</sup>, Maciej Lewenstein<sup>2,28</sup> and Alexandre Dauphin<sup>2</sup>

1 Faculty of Physics, University of Warsaw, Poland

2 ICFO - Institut de Ciències Fotòniques, The Barcelona Institute of Science and Technology,

08860 Castelldefels (Barcelona), Spain

3 Department of Physics, University of Basel, Switzerland

4 Quantum Technology Research Group, Heinrich-Heine-Universität Düsseldorf, Germany

6 Machine Learning Group, Technische Universität Berlin, Germany

8 Department of Applied Physics, Aalto University, Espoo, Finland

9 Institute of Physics, Albert-Ludwig University of Freiburg, Germany

10 International Centre for Theory of Quantum Technologies, University of Gdańsk, Poland

11 Department of Algorithms and System Modeling, Faculty of Electronics, Faculty of Electronics,

Telecommunications and Informatics, Gdańsk University of Technology, Poland

12 Institute for Theoretical Physics, University of Innsbruck, Austria

13 Department of Chemistry, University of Toronto, Canada 14 Vector Institute for Artificial Intelligence, MaRS Centre, Toronto, Canada 15 Barcelona Supercomputing Center, Spain 16 LIACS Leiden University The Netherlands 17 CMAP, École Polytechnique, France 18 Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland 19 Menten AI, Inc., Palo Alto, California, United States of America 20 Niels Bohr Institute, Copenhagen, Denmark 21 Beijing National Lab for Condensed Matter Physics

and Institute of Physics, Chinese Academy of Sciences, Beijing, China 22 Songshan Lake Materials Laboratory, Dongguan, China 23 Perimeter Institute for Theoretical Physics, Waterloo, Canada

24 Kavli Institute of Nanoscience, Delft University of Technology, NL-2600 GA Delft, The Netherlands 25 Department of Chemistry, University of British Columbia, Vancouver, Canada 26 Max Planck Institute for the Science of Light, Erlangen, Germany

27 Department of Physics, Friedrich-Alexander Universität Erlangen-Nürnberg, Germany 28 ICREA, Pg. Lluís Companys 23, 08010 Barcelona, Spain These authors contributed equally,

> \* Anna.Dawid@fuw.edu.pl, Alexandre.Dauphin@icfo.eu June 23, 2022

In these Lecture Notes, we provide a comprehensive introduction to the most recent advances in the application of machine learning methods in quantum sciences. We cover the use of deep learning and kernel methods in supervised, unsupervised, and reinforcement learning algorithms for phase classification, representation of many-body quantum states, quantum feedback control, and quantum circuits optimization. Moreover, we introduce and discuss more specialized topics such as differentiable programming, gener-

ative models, statistical approach to machine learning, and quantum machine learning.

Abstract

5 Université de Paris, CNRS, Laboratoire, Matériaux et Phénomènes Quantiques, France

7 BIFOLD, Berlin Institute for the Foundations of Learning and Data, 10587 Berlin, Germa

- <sup>2</sup>Rigetti Computing, 2919 Seventh Street, Berkeley, CA 94710
- 2 3 Wallenberg Centre for Quantum Technology, Department of Microtechnology and Nanoscience, Chalmers University of Technology, 412 96 Gothenburg, Sweden
- Feb <sup>4</sup>data cybernetics, Martin-Kolmsperger-Str 26, 86899 Landsberg, Germany
- 4 <sup>5</sup>Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany

he framework for optimization and machine learning of quantum and hybrid quantum ry provides a unified architecture for near-term quantum computing devices, supporting le paradigms. PennyLane's core feature is the ability to compute gradients of variational compatible with classical techniques such as backpropagation. PennyLane thus extends rithms common in optimization and machine learning to include quantum and hybrid akes the framework compatible with any gate-based quantum simulator or hardware. We ields, Rigetti Forest, Qiskit, Cirq, and ProjectQ, allowing PennyLane optimizations to be im devices provided by Rigetti and IRM O. On the classical front. PennyLane interfaces ig libraries such as TensorFlow PyTorch, and autograd. Pennyl ane can be used for the im eigensolvers, quantum approximate optimization, quantum machine learning models

> tum computing with applications in quantum chemistry [1] quantum optimization [2], factoring [3], state diagonaliza-tion [4], and quantum machine learning [5–18]. In a reversal from the usual practices in quantum computing re

ment and commercializasearch, a lot of research for these mostly heuristic algo s had a profound impact rithms necessarily focuses on numerical experiments rathe algorithms. Near-term than rigorous mathematical analysis. Luckily, there are vari ies that are of shallow ous publicly accessible platforms to simulate quantum algo The design paradigm of rithms [19-26] or even run them on real quantum device e quantum and dassical through a cloud service [27, 28]. However, even though increasingly important. some frameworks are designed with variational circuits in ss of hybrid algorithms is mind [25, 29, 30], there is at this stage no unified tool for are parameter-dependent the hybrid optimization of quantum circuits across quantum optimized by a classical platforms, treating all simulators and devices on the same objective. . footing. PennyLane is an open-source Python 3 framework that fa-

ational circuits opens up cilitates the optimization of quantum and hybrid quantum ues for near-term quanclassical algorithms. It extends several seminal ma

A Practical Guide to **Quantum Machine Learnina** and Quantum Optimization

Hands-on Approach to Modern Quantum Algorithms

SAMUEL GONZÁLEZ-CASTILLO

### ELÍAS F. COMBARRO Foreword by Alberto Di Mealio

Head of Innovation - Coordinator CERN Quantum Technology Initiative

## **Thank you!**

## **Any questions?**

### 

This presentation has been released under the Creative Commons CC BY-NC-ND license, i.e.

BY: credit must be given to the creator. NC: Only noncommercial uses of the work are permitted. ND: No derivatives or adaptations of the work are permitted.

Photos from Unsplash

Enquanted is being somewhere in-between Enchanted and Entangled