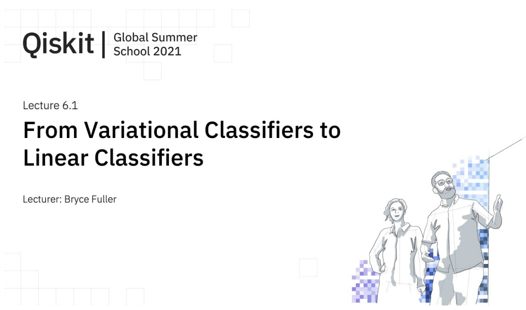
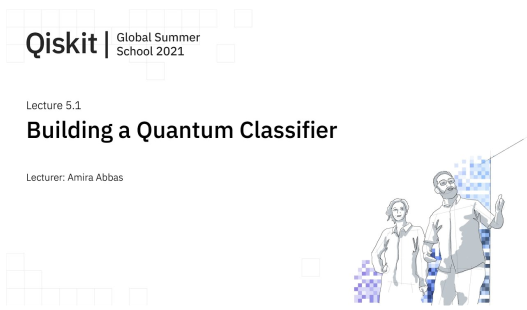
Project themes

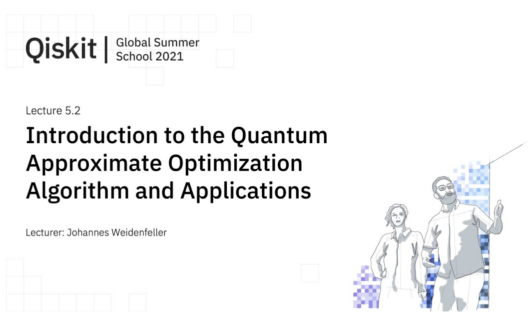
1. **Classification with variational quantum circuits**
   - *Tasks*:
     - Choose or prepare a dataset for training and testing.
     - Design a variational quantum circuit, ranging from the encoding of data, to ansatz, and the corresponding measurement and classical post-processing that converts the output of the quantum circuit in a classification.
     - Compare the classification obtained with a previously selected classical algorithm using a carefully selected metric.
   - *References*:
     - Machine Learning with Quantum Computers - Maria Schuld - Chapter 5
     - Supervised Learning with quantum.enhanced feature spaces - https://www.nature.com/articles/s41586-019-0980-2
     - Parameterized quantum circuits - https://learn.qiskit.org/course/machine-learning/parameterized-quantum-circuits
     - https://medium.com/qiskit/building-a-quantum-variational-classifier-using-real-world-data-809c59eb17c2
     - Qiskit tutorial - https://qiskit.org/documentation/stable/0.24/tutorials/machine_learning/03_vqc.html
     - Qiskit tutorial - https://learn.qiskit.org/course/machine-learning/variational-classification
     - https://qiskit.org/documentation/machine-learning/tutorials/02_neural_network_classifier_and_regressor.html
     - Variational classifier - https://pennylane.ai/qml/demos/tutorial_variational_classifier.html
     - Data-reuploading classifier - https://pennylane.ai/qml/demos/tutorial_data_reuploading_classifier.html





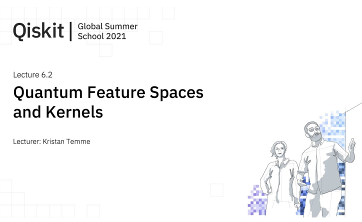
2. Solving Quadratic Unconstrained Binary Optimization problems (QUBO's) using QAOA

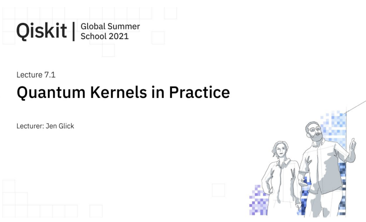
- *Tasks*:
  - Choose or formulate a QUBO instance for a problem of interest.
  - Encode the QUBO instance as a problem Hamiltonian.
  - Use the QAOA algorithm to solve the optimization problem i.e. the ground state of the problem Hamiltonian should be the solution to the problem.
  - Evaluate the quality of the solution obtained. Discuss or present alternatives to further improve the quality of the solution.
- *References*:
  - Machine Learning with Quantum Computers - Maria Schuld - Section 3.6.5
  - Ising formulations of many NP problems - https://arxiv.org/pdf/1302.5843.pdf
  - Quantum computation by adiabatic evolution - https://arxiv.org/abs/quant-ph/0001106
  - A Quantum Approximate Optimization Algorithm - https://arxiv.org/abs/1411.4028
  - From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz - https://arxiv.org/abs/1709.03489
  - Solving combinatorial optimization problems using QAOA - https://qiskit.org/textbook/ch-applications/qaoa.html
  - Quantum Approximate Optimization Algorithm - https://qiskit.org/documentation/tutorials/algorithms/05_qaoa.html
  - Introduction to QAOA - https://pennylane.ai/qml/demos/tutorial_qaoa_intro.html
  - QAOA for MaxCut - https://pennylane.ai/qml/demos/tutorial_qaoa_maxcut.html
  - Warm start optimization - https://qiskit.org/ecosystem/optimization/tutorials/10_warm_start_qaoa.html

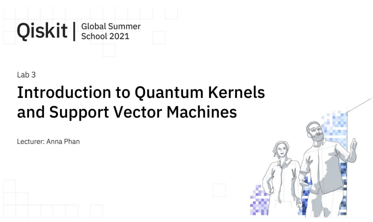


3. Support Vector Machines (SVM) with quantum kernels

- *Tasks:*
  - Choose or prepare a dataset for training and testing.
  - Design a data encoding strategy for the dataset. *Pay attention that kernelized methods don't use an ansatz. Thus, all possible quantum enhancements comes from the encoding itself.*
  - Evaluate the quality of the classification obtained from the quantum kernel algorithm. The analysis can be empirical or theoretical. For instance, compare the accuracy of the quantum kernel algorithm with a classical one or compare the quantum and classical algorithm in terms of memory i.e. the number of qubits, the number of features …
- *References:*
  - Machine Learning with Quantum Computers - Maria Schuld - Chapter 6
  - Supervised Learning with quantum.enhanced feature spaces - https://www.nature.com/articles/s41586-019-0980-2
  - Quantum machine learning in feature Hilbert spaces - https://arxiv.org/abs/1803.07128
  - Supervised quantum machine learning models are kernel methods - https://arxiv.org/abs/2101.11020
  - Training Quantum Embedding Kernels on Near-Term Quantum Computers - https://arxiv.org/abs/2105.02276
  - Quantum feature maps and kernels - https://learn.qiskit.org/course/machine-learning/quantum-feature-maps-kernels
  - Kernel-based training of quantum models - https://pennylane.ai/qml/demos/tutorial_kernel_based_training.html
  - Pegasus quantum support vector classifier - https://qiskit.org/documentation/machine-learning/tutorials/07_pegasos_qsvc.html
  - Training and evaluating quantum kernels - https://pennylane.ai/qml/demos/tutorial_kernels_module.html

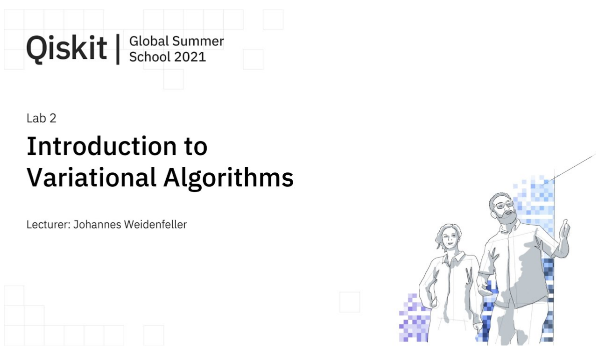






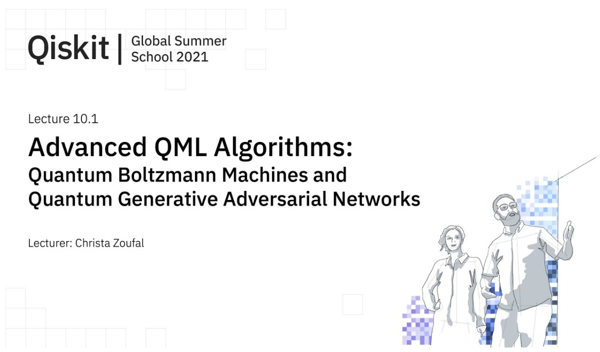
5. Quantum Data Encoding and Variational Circuits: a performance analysis

- *Tasks*:
  - Select or prepare a dataset to be analyzed.
  - Select different encoding strategies to use in a previously selected variational classifier. The students can abstract from the ansatz I.e. a well known ansatz can be used since the focus of this work will be the encoding of data.
  - Evaluate the impact of data encoding in the performance of the variational quantum classifier. The capacity of the encoding can be assessed trough a number of ways, for instance, comparing the accuracy of the classifier in terms of the number of iterations of the algorithm, the memory used I.e. the number of qubits, the number of repetitions of the encoding …
- *References*:
  - Machine Learning with Quantum Computers - Maria Schuld - Chapter 4
  - Robust Data encoding for quantum classifiers - https://arxiv.org/abs/2003.01695
  - Quantum embeddings for machine learning - https://arxiv.org/abs/2001.03622
  - The effect of data encoding on the expressive power of variational quantum machine learning models - https://arxiv.org/abs/2008.08605
  - Supervised learning with quantum enhanced feature spaces - https://arxiv.org/abs/1804.11326
  - Data encoding - https://learn.qiskit.org/course/machine-learning/data-encoding



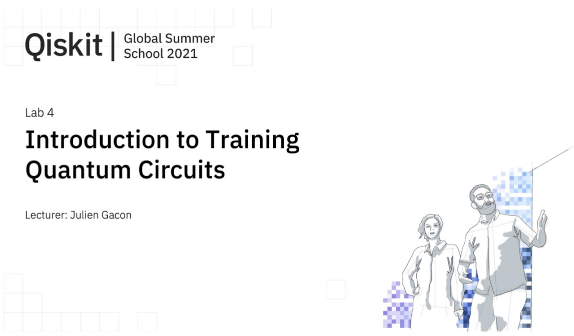
6. Preparing distributions with Quantum Generative Adversarial Networks

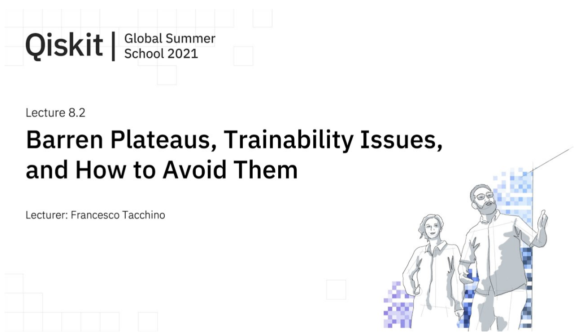
- *Summary*: Preparing a probability distribution in a quantum computer is generally hard. However, quantum Generative Adversarial Networks (qGAN's) can learn a distributions from samples instead of the full distribution. This strategy could be used to reduce the complexity of preparing distributions in the quantum setting.
- *Tasks*:
  - Select or prepare a classical discrete probability distribution.
  - Generate samples from the distribution.
  - Design the variational qGAN to learn the probability distribution based on the generated samples.
  - Evaluate the capacity of the qGAN. The capacity can be evaluated using different strategies. For instance, the distance between the generated probability distribution and the true probability distribution can be used as a metric (see Kullback-Leibler divergence) . Another strategy is to compare the number of gates used to encode the probability distribution compared to full amplitude encoding. Or compare the accuracy in terms of the data samples using in training.
- *References*:
  - Machine Learning with Quantum Computers - Maria Schuld - Chapter 5.3.4
  - Quantum generative adversarial learning - https://arxiv.org/pdf/1804.09139.pdf
  - Quantum generative adversarial networks - https://arxiv.org/pdf/1804.08641.pdf
  - qGANs for Loading Random Distributions - https://qiskit.org/documentation/machine-learning/tutorials/04_qgans_for_loading_random_distributions.html
  - Unsupervised Learning - https://learn.qiskit.org/course/machine-learning/unsupervised-learning
  - Quantum GAN's - https://pennylane.ai/qml/demos/tutorial_quantum_gans.html



7. Optimizing the optimizer: Discovering the best optimization strategy

- *Tasks*:
  - Select or prepare a dataset for training and testing.
  - Implement a well known variational quantum classifier.
  - Use different training strategies. For instance, learning rates, initialization, and most importantly optimizers.
  - Evaluate the performance of the variational quantum classifier for such training strategies. The performance should be assessed though the lens of accuracy, trainability and convergence.
- *References*:
  - Machine Learning with Quantum Computers - Maria Schuld - Chapter 5.3
  - Barren plateaus in quantum neural network training landscapes - https://www.nature.com/articles/s41467-018-07090-4
  - Quantum Natural Gradients - https://arxiv.org/abs/1909.02108
  - Quantum Natural Gradients - https://pennylane.ai/qml/demos/tutorial_quantum_natural_gradient.html
  - Rotosolve - https://arxiv.org/pdf/1905.09692.pdf
  - Training parameterized quantum circuits - https://learn.qiskit.org/course/machine-learning/training-quantum-circuits





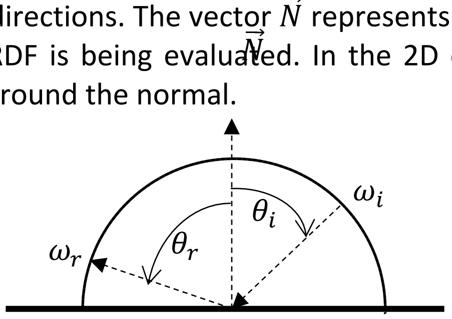
8. Variational Quantum Regression applied to Computer Graphics

Rendering is the branch of Computer Graphics concerned with the synthesis of images using computers. A fundamental factor in order to render highly realistic images (such as Figure 1) is to have very good characterizations of the materials that constitute the objects in the scene. A material's characterization specifies what fraction of the light incident on that material along a given direction is reflected along some other direction. The function that characterizes a material is referred to as the **B**idirectional **R**eflectance **D**istribution **F**unction (BRDF).

The goal of this project is to learn a BRDF using a Variational Quantum Regression approach.

Figure 1-Computer rendered image

The BRDF returns which fraction of the radiance incident along direction ω_i is reflected along direction ω_r. The figure below presents the BRDF schematically in a 2D plane for a particular pair of directions. The vector $\vec{N}$ represents the normal to the surface at the point where the BRDF is being evaluated. In the 2D case the BRDF is defined in the semicircle defined around the normal.



We are interested in. the 3D case as depicted in Figure 2.

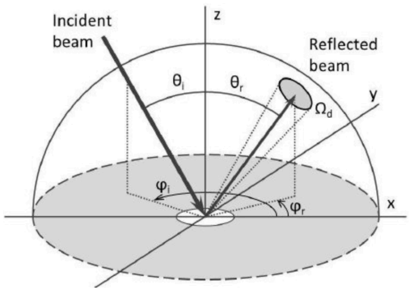


Figure 2-the BRDF in the hemisphere

Note that each direction is now given by two angles, $\omega = (\theta, \varphi)$, with $\theta \in \left[0, \frac{\pi}{2}\right]$ being the elevation angle and $\varphi \in [0,2\pi]$ the orientation angle. A local coordinate system is used with the Z axis aligned with the normal and the X axis serving as the reference for the orientation angle.

The BRDF is therefore a function of 2 directions and represented as $f_r(\omega_i, \omega_r)$ or equivalently $f_r(\theta_i, \varphi_i, \theta_r, \varphi_r)$.

There are well known analytical models for BRDFs, the simplest ones being given in Table 1. Realistic BRDFs are also available as tables, parameterized by the 4 angles.

Table 1-Simple analytic BRDFs

BRDF	$f_r(\theta_i, \varphi_i, \theta_r, \varphi_r)$
Lambert	$k_d * \cos(\theta_i)$
Phong	$k_d * \cos(\theta_i) + k_s * \left(\cos\left(\omega_{i\_ref}, \omega_r\right)\right)^n$ with $\cos\left(\omega_{i\_ref}, \omega_r\right) = \sin\theta_i \sin\theta_r \cos(\varphi_i - \varphi_r) - \cos\theta_i \cos\theta_r$

The aim of this project is to develop and assess a Variational Quantum Regression approach which can learn BRDFs. The resulting model should be able to estimate the BRDF value given a pair of directions (ω_i, ω_r).

References

https://en.wikipedia.org/wiki/Bidirectional_reflectance_distribution_function

https://www.youtube.com/watch?v=bv4Rvdc7-YM