Quantum Machine Learning

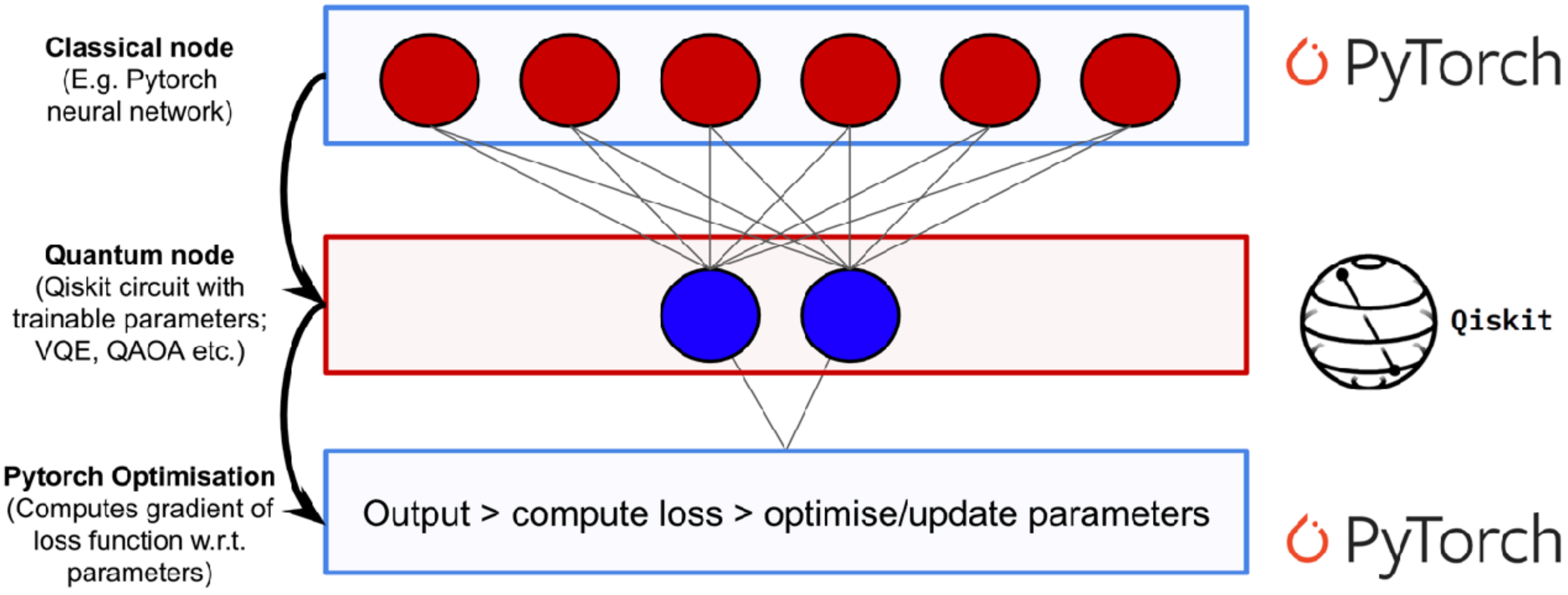
Hung-Wei Tseng

https://github.com/qiskit-community/qiskit-machine-learning/tree/main

Machine Learning (ML)

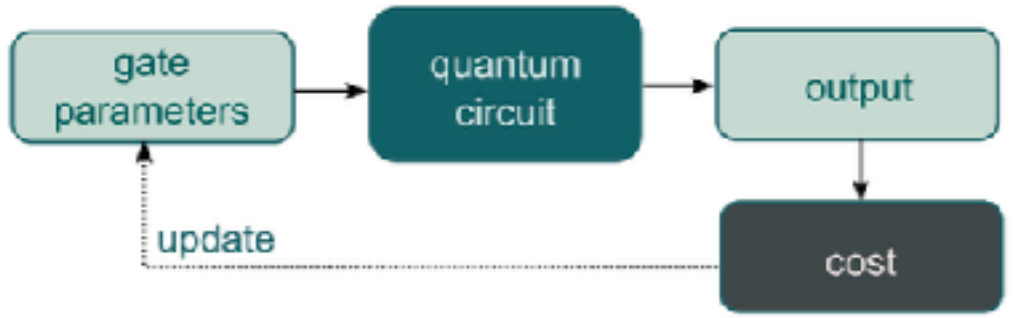
- Supervised learning — given tuples of labeled data (x_i, y_i), we aim to learn the function that maps $f: x \mapsto y$ and generalizes to unseen inputs. e.g., given a set of labeled photos of cats or dogs, we want to identify new photos of cats or dogs.
- Unsupervised learning — given a collection of unlabeled data (x_i), we aim to learn some structure of the data. e.g., grouping a set of viewers based on their movie viewing history in order to recommend new movies.
- Reinforcement learning — given access to an environment that rewards us based on our actions, we aim to maximize our expected rewards; for example, algorithmically learning how to play PAC-MAN.

Hybrid quantum-classical Neural Networks

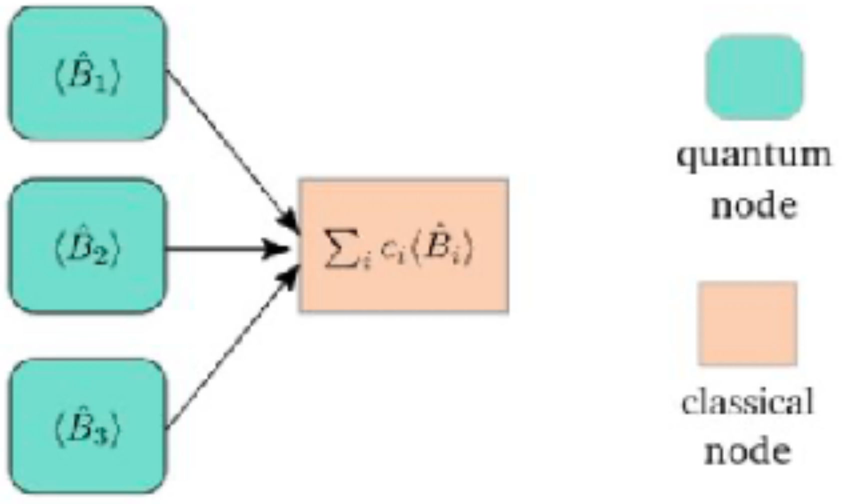


Hybrid Computation

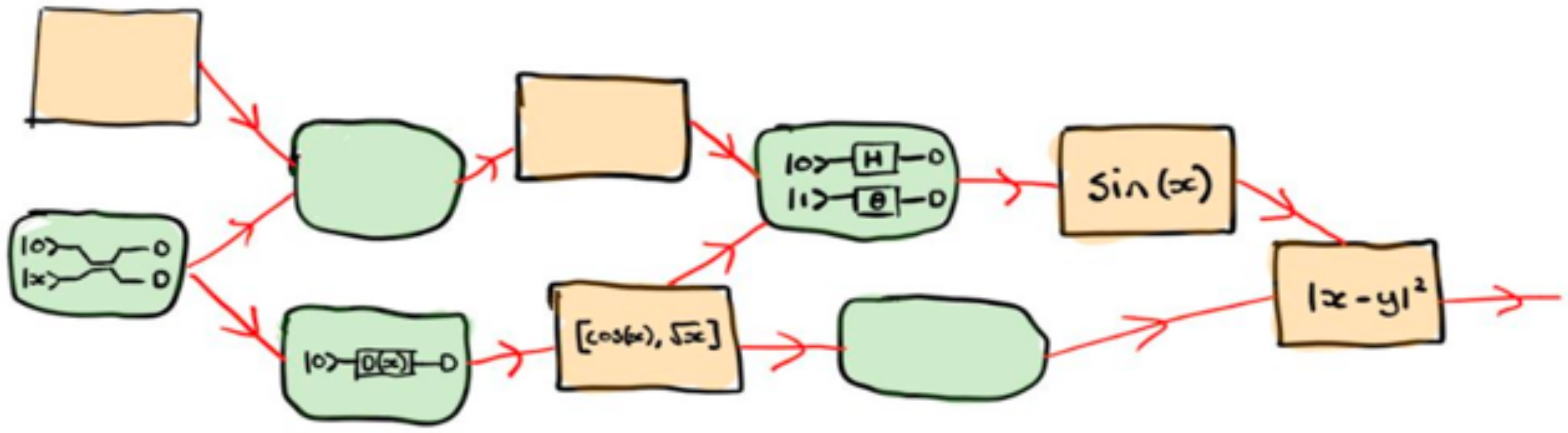
- Use QPU with classical coprocessor
  - Classical optimization loop



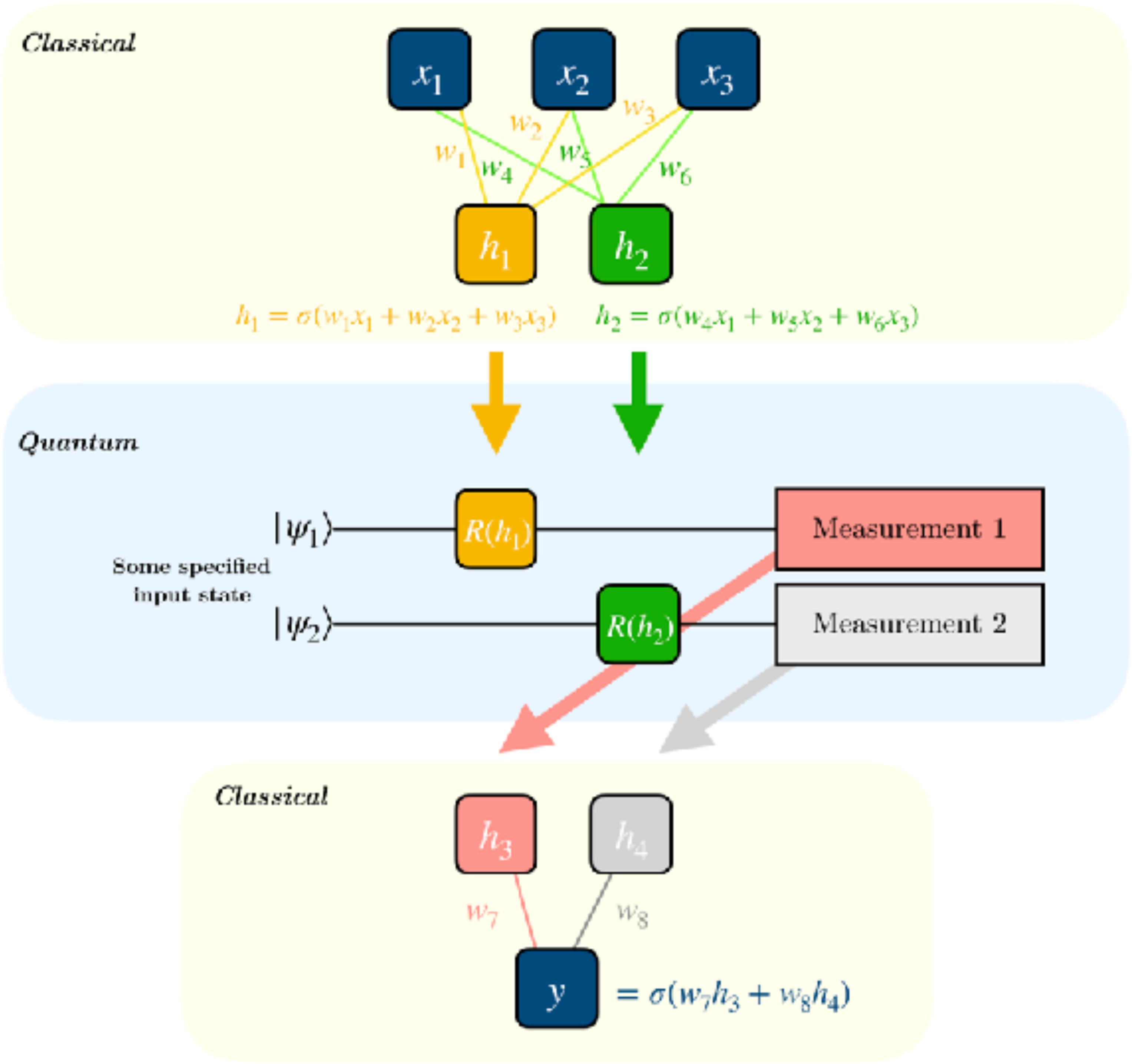
  - Pre-/post-process quantum circuit outputs



  - Arbitrarily structured hybrid computations



An exemplary implementation



Classical data using Quantum algorithms

- We're aiming at processing Classical data using Quantum machine learning algorithms
- To obtain quantum advantage, the Kernel must be very Hard
- This speedup is enabled by
  - Quantum phase estimation
  - Quantum Fourier transform
  - Quantum simulation methods
  - Matrix multiplication — A Quantum gate is a linear layer of a giant neural net !

Parameterized quantum circuits — the core component of QML

Parameterized quantum circuits

- Gates are defined through tunable parameters
- Parameterized circuit itself can be described as a unitary operation on n qubits, $\mathbf{U}_\theta$, $|\psi_\theta\rangle = \mathbf{U}_\theta|\phi_0\rangle$, where θ is a set of tunable parameters.
- To encode data, where the parameters are determined by the data being encoded
- As a quantum model, where the parameters are determined by an optimization process.

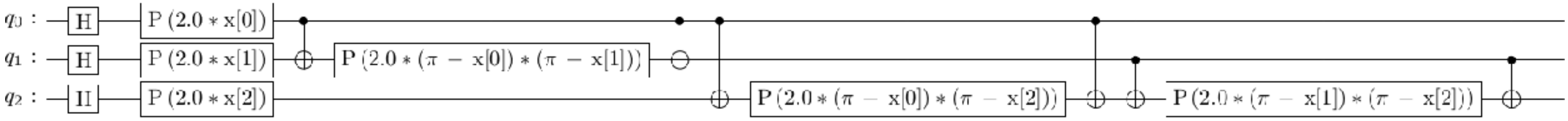
Properties of Parameterized quantum circuits

- Expressibility
- Entangling capability
- Hardware efficiency

Exemplary PQCs — ZZFeatureMap

- Data encoding

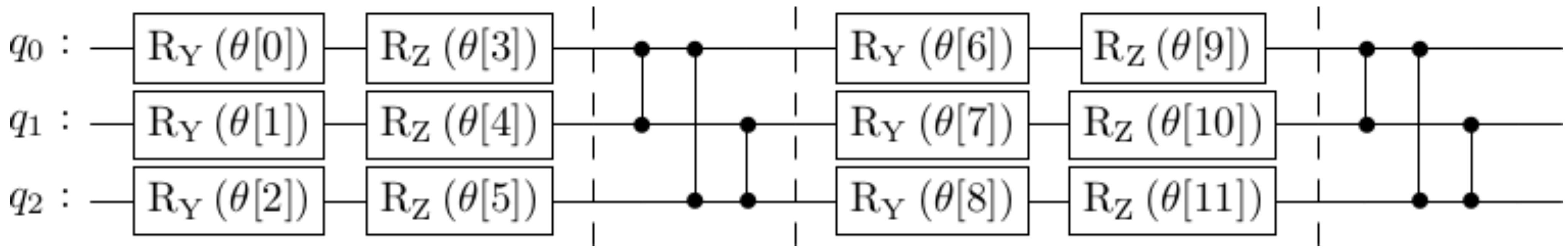
$$\mathcal{U}_{\Phi(\mathbf{x})} = \prod_d U_{\Phi(\mathbf{x})} H^{\otimes n}, \; U_{\Phi(\mathbf{x})} = \exp\left(i \sum_{S \subseteq [n]} \phi_S(\mathbf{x}) \prod_{k \in S} P_i \right)$$



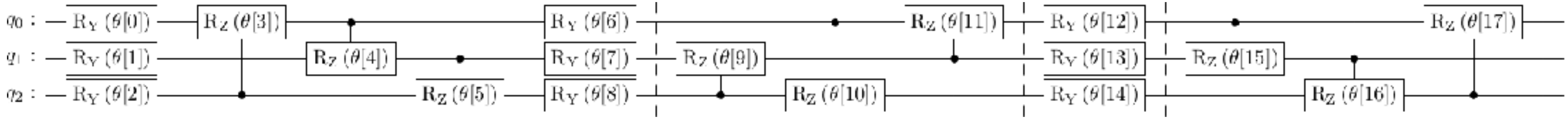
Vojtech Havlicek, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow and Jay M. Gambetta, Supervised learning with quantum enhanced feature spaces, Nature 567, 209-212 (2019), doi.org:10.1038/s41586-019-0980-2, arXiv:1804.11326.

Exemplary PQCs (cont.)

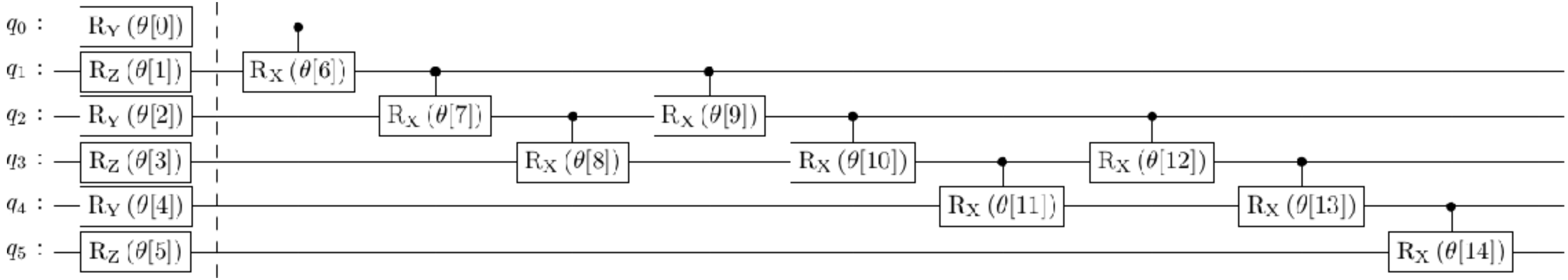
- Quantum models



- Vojtech Havlicek, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow and Jay M. Gambetta, Supervised learning with quantum enhanced feature spaces, Nature 567, 209-212 (2019), doi.org:10.1038/s41586-019-0980-2, arXiv:1804.11326.



- Sukin Sim, Peter D. Johnson and Alan Aspuru-Guzik, Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms, Advanced Quantum Technology 2 (2019) 1900070, doi:10.1002/qute.201900070, arXiv:1905.10876.



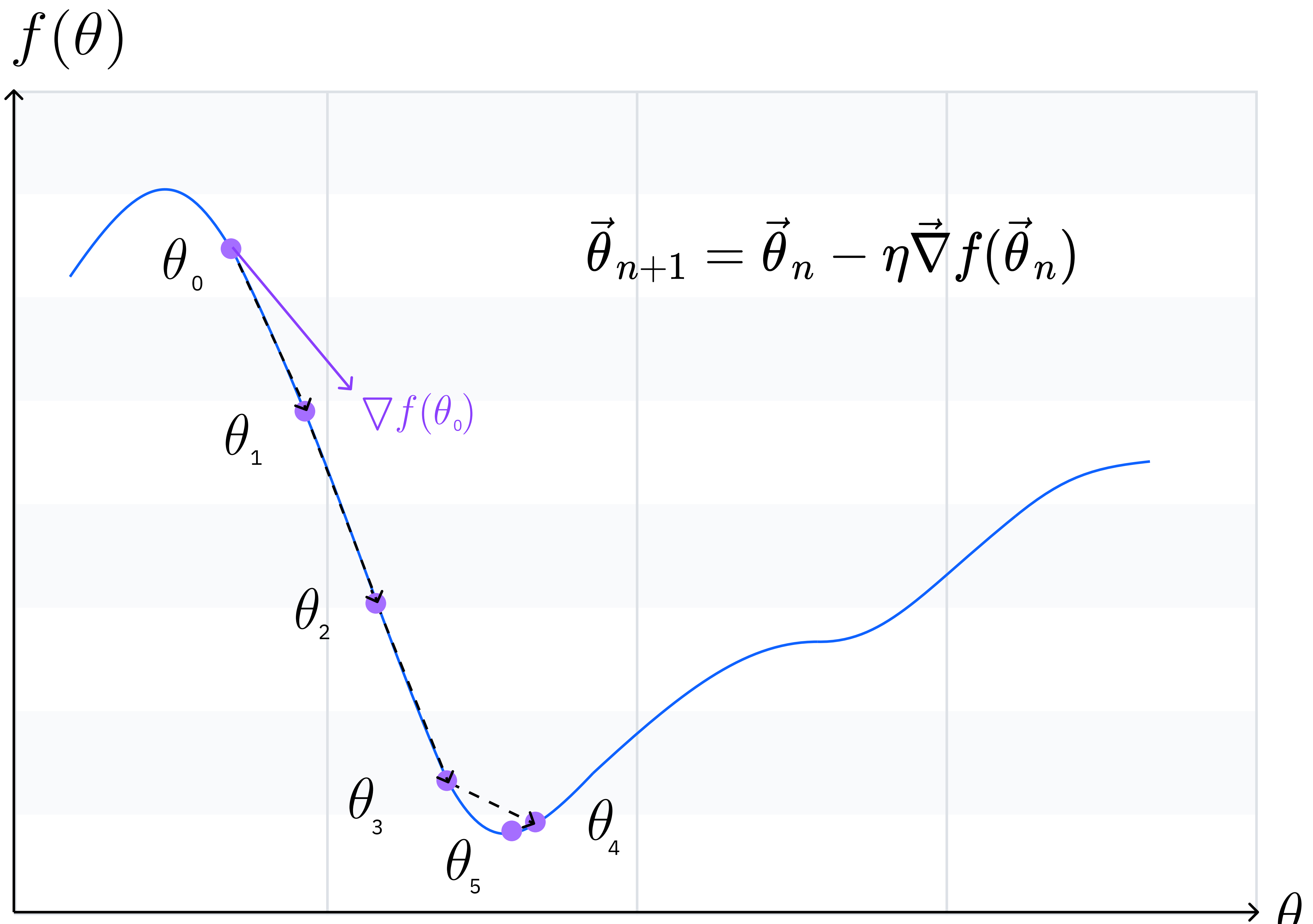
-

Training parameterized quantum circuits

How to ‘train’ quantum circuits?

- Simulator-based
  - Build simulation inside existing classical library
  - Can leverage existing optimization & ML tools
  - Great for small circuits, but not scalable
- Hardware-based
  - No access to quantum information; only have measurements & expectation values
  - Needs to work as hardware becomes more powerful and cannot be simulated

Gradients



$f(\theta)$
$\vec{\theta}_{n+1} = \vec{\theta}_n - \eta\vec{\nabla} f(\vec{\theta}_n)$
θ_0
$\nabla f(\theta_0)$
θ_1
θ_2
θ_3
θ_4
θ_5
θ

Gradients of quantum circuits

- Training strategy: use gradient descent algorithms.
- Need to compute gradients of variational circuit outputs w.r.t. their free parameters.
- How can we compute gradients of quantum circuits when even simulating their output is classically intractable?

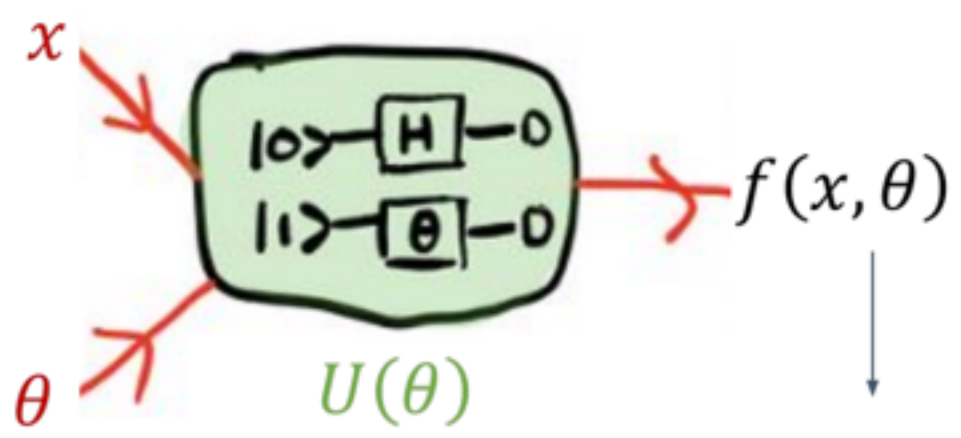
Quantum Circuit Learning

- Use the same device to compute a function and its gradient
- "Parameter shift" differentiation rule: gives exact gradients
- Minimal overhead to compute gradients vs. original circuit
- Optimize circuits using gradient descent
- Compatible with classical backpropagation: hybrid models are end-to-end differentiable

$$f(\theta) = \sin\theta \quad \Rightarrow \quad \partial_\theta f(\theta) = \cos\theta$$

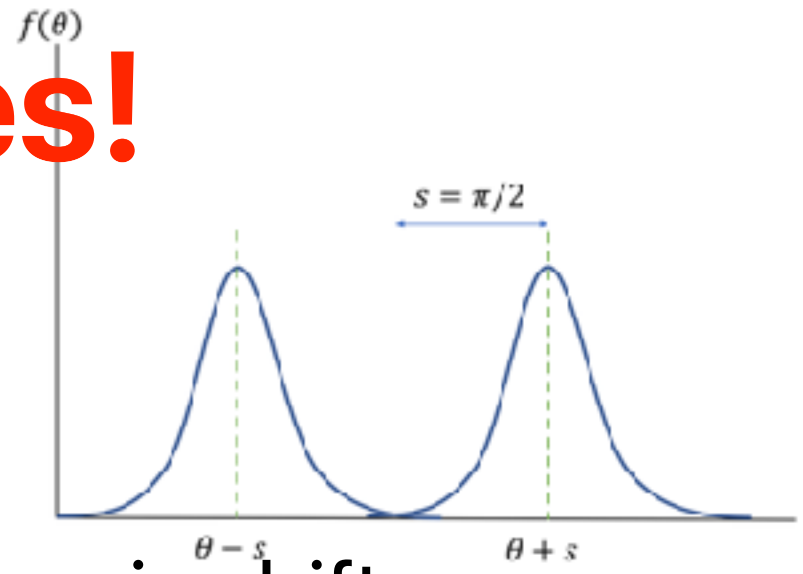
$$\cos\theta = \frac{\sin\left(\theta + \frac{\pi}{4}\right) - \sin\left(\theta - \frac{\pi}{4}\right)}{\sqrt{2}}$$

$$\partial_\theta f = \frac{1}{\sqrt{2}}\left(f\left(\theta + \frac{\pi}{4}\right) - f\left(\theta - \frac{\pi}{4}\right)\right)$$

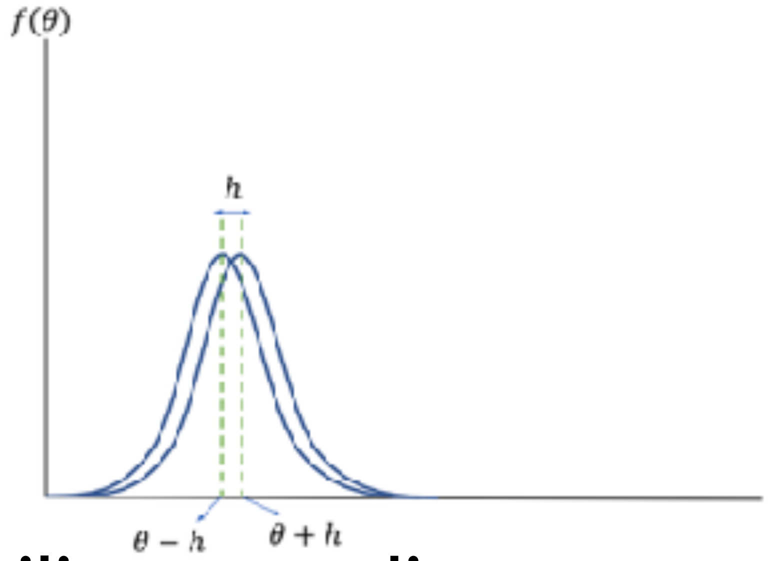


$$\partial_\theta f(\theta) = c[f(\theta + s) - f(\theta - s)]$$

This is not finite differences!



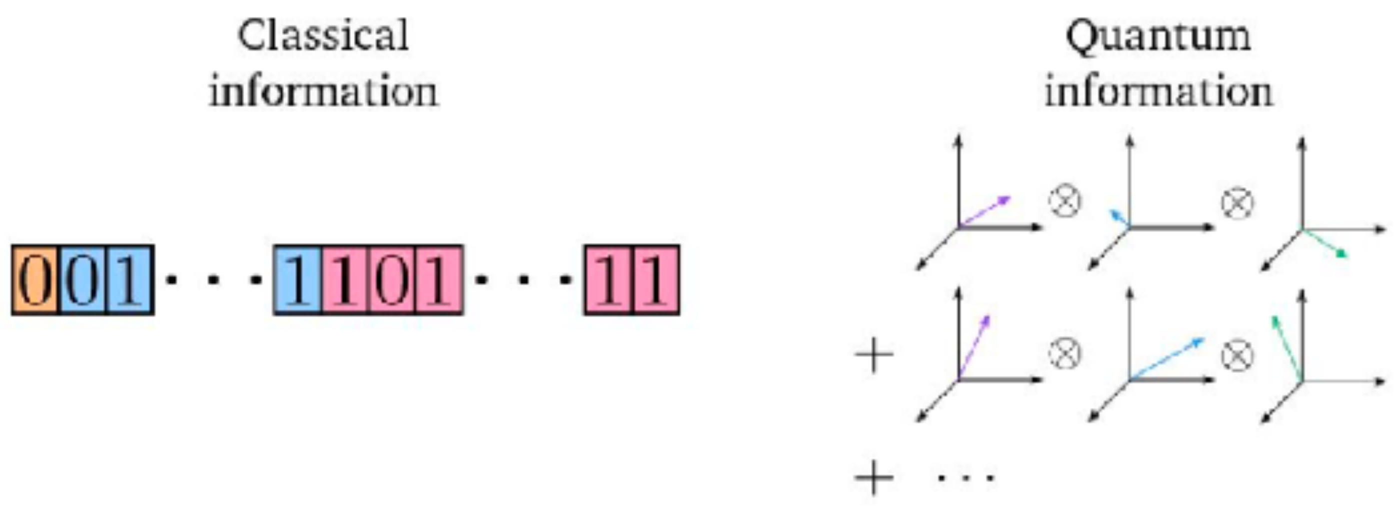
- $\partial_\theta f(\theta) = c[f(\theta + s) - f(\theta - s)]$
  - Exact
  - No restriction on the shift – in general, we want a macroscopic shift

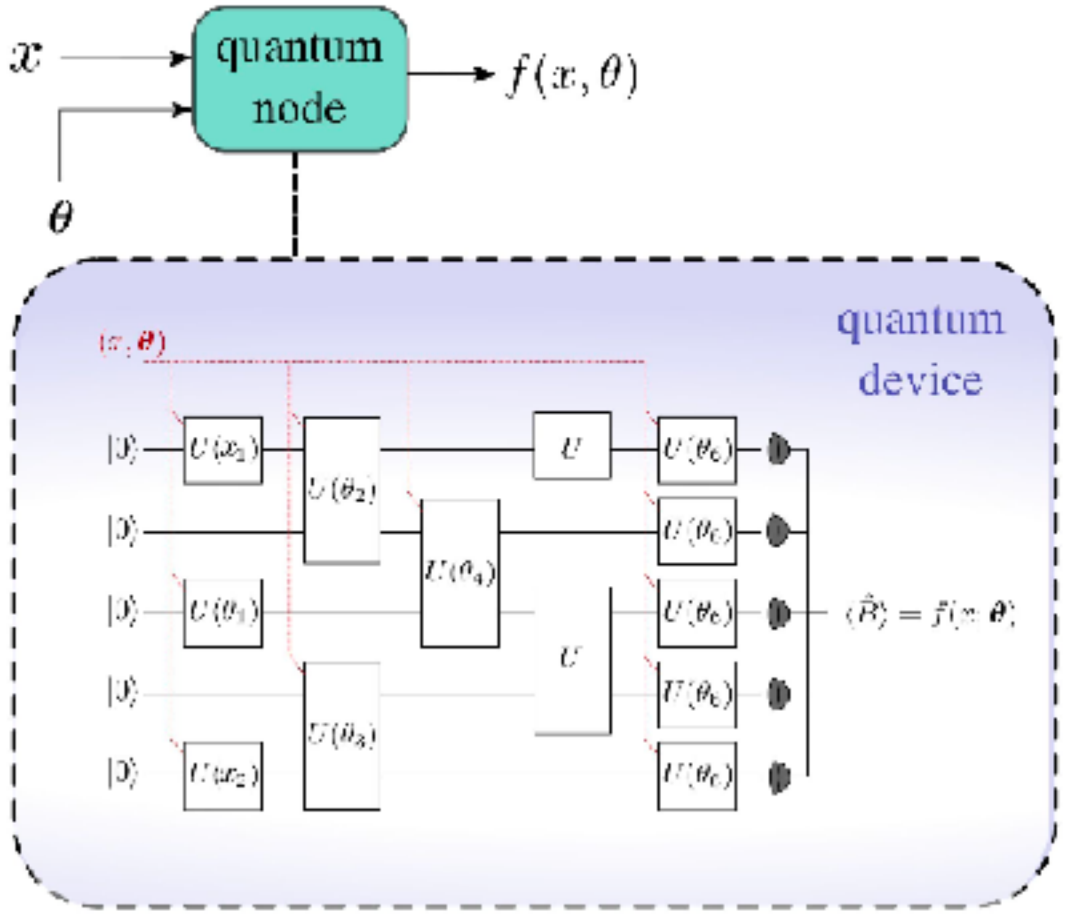


- $$\partial_\theta f(\theta) \approx \frac{f(\theta + h) - f(\theta - h)}{2h}$$
  - Only an approximation
  - Requires that h is small
  - In subject to the quirks of numerical differentiation — stability, rounding error, truncation error
  - For NISQ devices, small h could lead to the difference being swamped by noise

Quantum Nodes

- Classical and quantum information are distinct
- QNode: common interface for quantum and classical devices
  - Classical device sees a callable parameterized function
  - Quantum device sees fine-grained circuit details



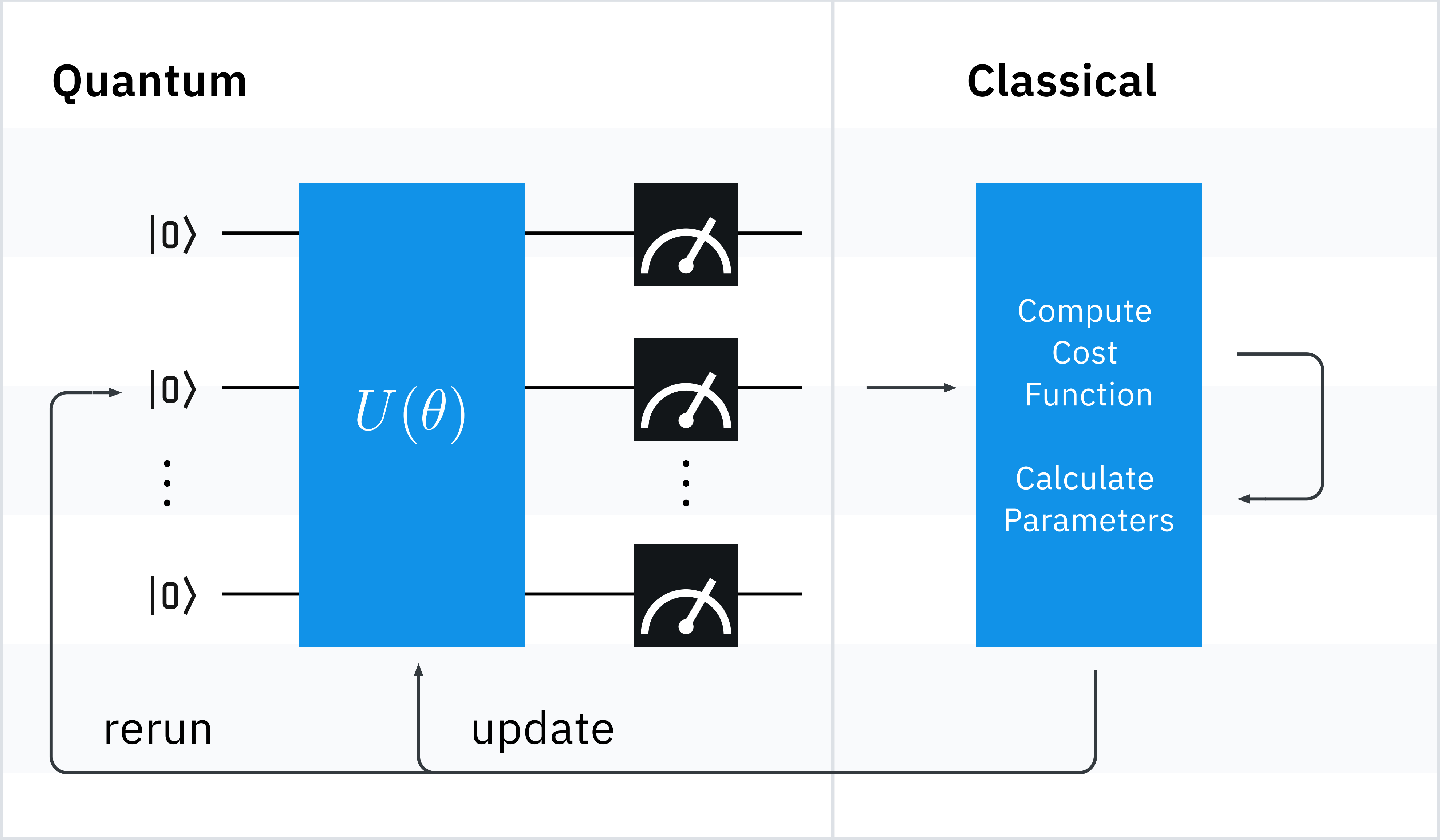


Supervised learning

- Classification — assign data into specific categories. e.g., given a set of labeled images of chairs or tables, try to identify new photos of chairs or tables
- Regression — understand the relationship between dependent and independent variables. It's commonly used to make predictions, e.g. given a series of historical stock prices, predict the future stock price

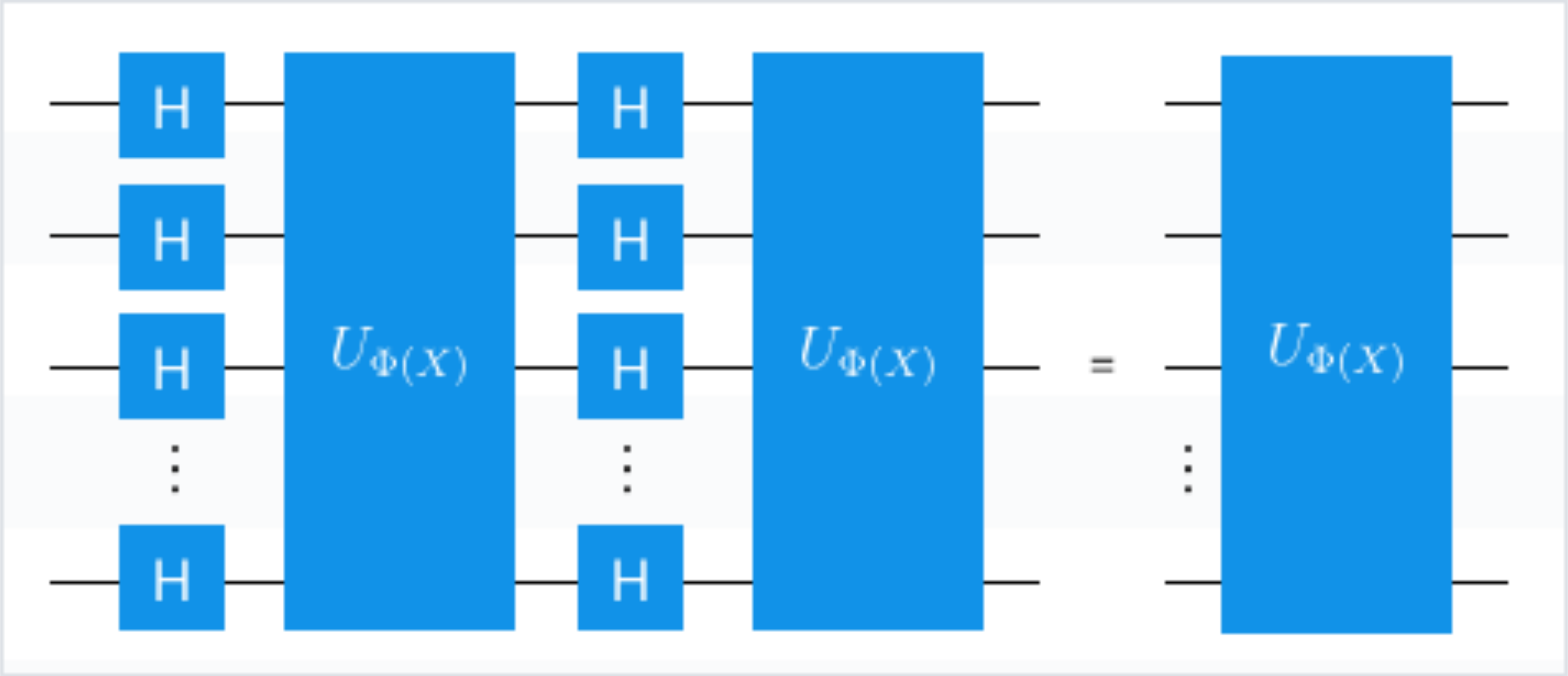
Quantum variational classification

- Use variational Circuit to generate a separating hyperplane in the quantum feature space - very much like linear binary classifier



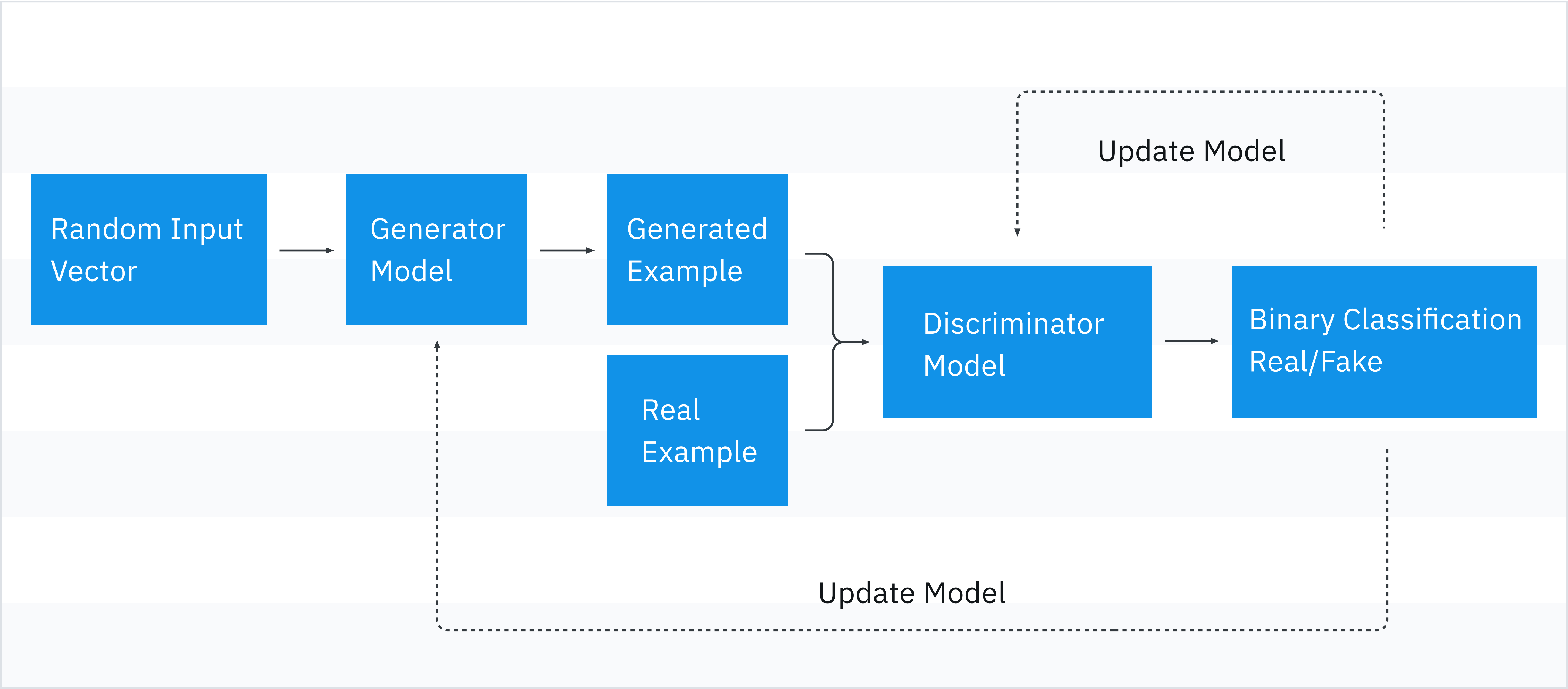
Quantum kernel estimation

- Use quantum computer to estimate the kernel function of the quantum feature space directly



$$\mathcal{U}_{\Phi(\vec{x})} = \prod_d U_{\Phi(\vec{x})} H^{\otimes n}, \; U_{\Phi(\vec{x})} = \exp\left(i \sum_{S \subseteq [n]} \phi_S(\vec{x}) \prod_{k \in S} P_i \right)$$

Quantum generative adversarial networks



$$\min_G \max_D \vec{E}_{x \sim P_{\mathrm{real}}} [\log D(x)] + \vec{E}_{z \sim P_G} [\log(1 - D(G(z)))]$$

Start the presentation to see live content. Still no live content? Install the app or get help at **PollEv.com/app**

Paper presentations

- Assignment #1 due this evening
- 3/11/2025 Haotian Lu — Hanrui Wang, Zirui Li, Jiaqi Gu, Yongshan Ding, David Z. Pan, and Song Han. OC: quantum on-chip training with parameter shift and gradient pruning. In the 59th ACM/IEEE Design Automation Conference (DAC '22)
- 3/11/2025
- 3/13/2025
- 3/13/2025
- Topics
  - Quantum optimization algorithms
  - Quantum compilers
  - Quantum architectures/memory