Ciência de Dados Quântica 2021/22

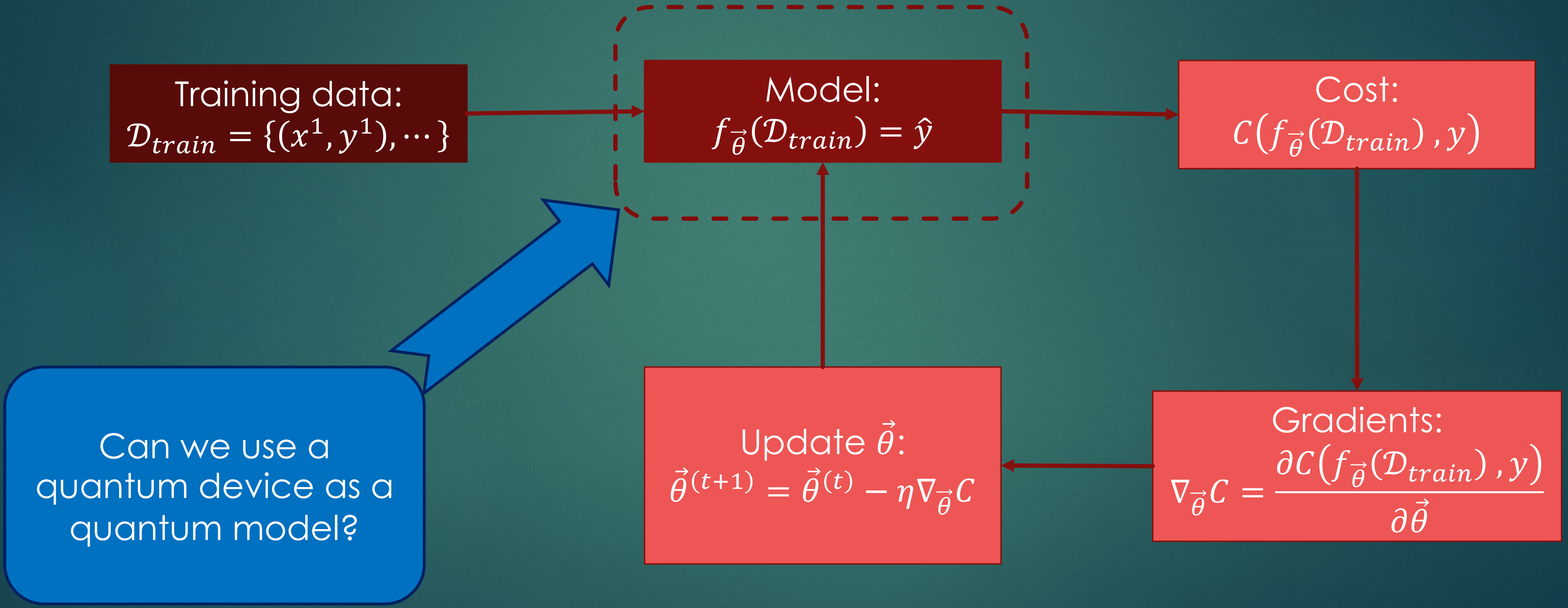
Variational Quantum Classification: an overview

LUÍS PAULO SANTOS

Material de Consulta

- [Schuld2021] – Chap. 5
- [Abbas2021] – 2021 Qiskit Global Summer School on Quantum Machine Learning:
  Building a Quantum Classifier
  https://learn.qiskit.org/summer-school/2021/lec5-1-building-quantum-classifier

General framework

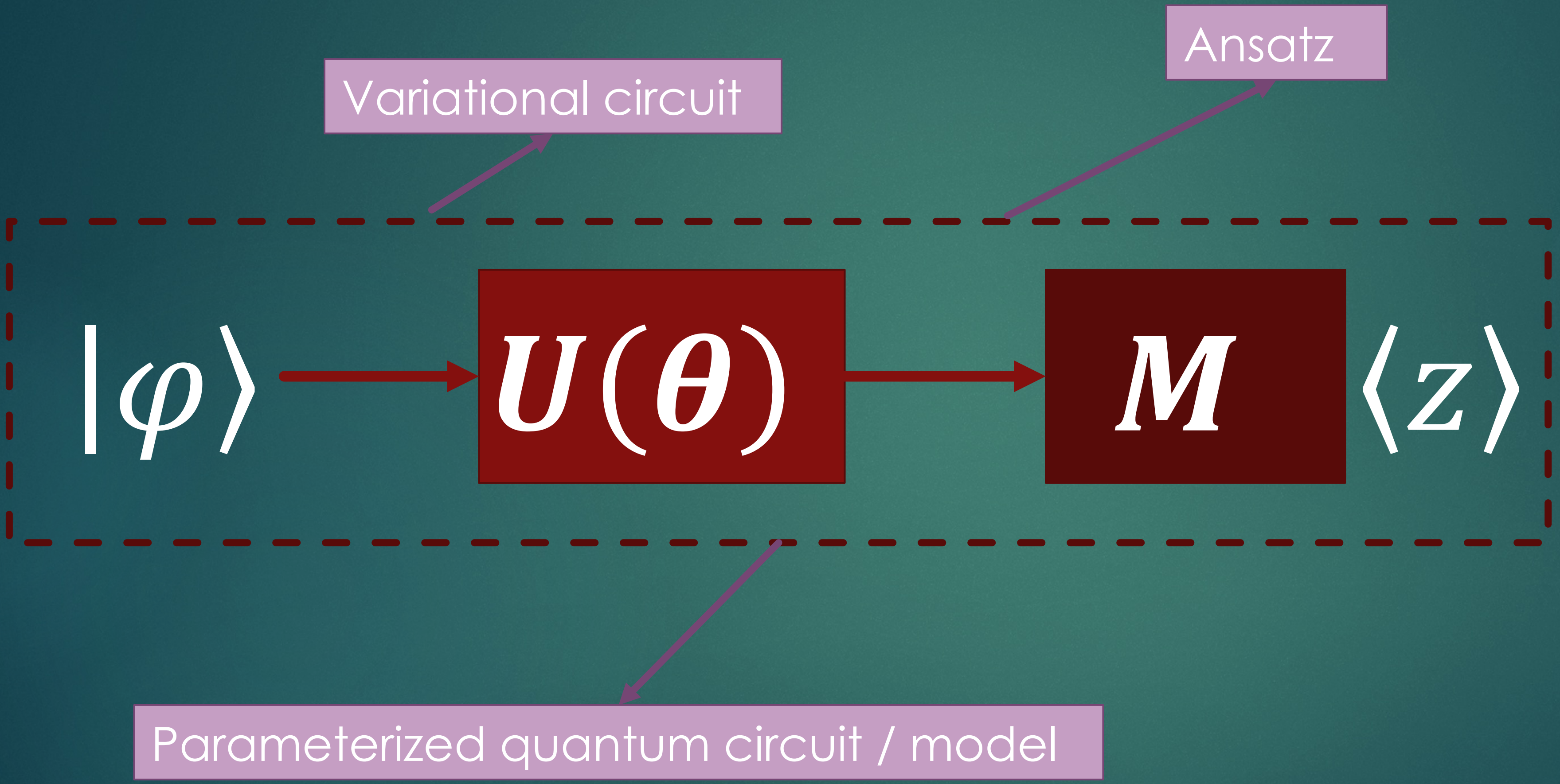


Near Term versus Fault Tolerant

- **Fault Tolerant** :
  - Large number of qubits, error corrected, noise resilient
  - Large and long programs, deep circuits
- **NISQ** : **N**oisy **I**ntermediate **S**cale **Q**uantum systems
  - Limited number of qubits, noise sensitive, decoherence
  - Short lived programs, shallow circuits

[Abbas2021]

Variational Models



- The output is stochastic
- Multiple measurements
- Expectation value
- Probability distribution over basis states

Variational circuit as a classifier

- **Task:**
  Train a parameterized quantum circuit on labelled samples for a set of classical data
  in order to predict label for new, unseen, data

1. **Encode the classical data into a quantum state**
2. **Apply a parameterized model**
3. **Measure the circuit to extract labels**
4. **Use optimization to update the model's parameters**

Data encoding

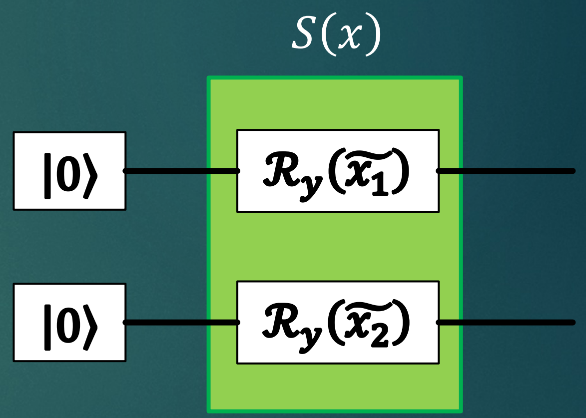
- $|\varphi\rangle$ is the quantum state encoding some classical data point, x^i, which is eventually a vector of multiple features: $x^i = \left(x_1^i, \cdots, x_N^i\right)$
- This encoding should be made explicit : $U(\theta)|\varphi\rangle = U'(x,\theta)|0\rangle = W(\theta)S(x)|0\rangle$

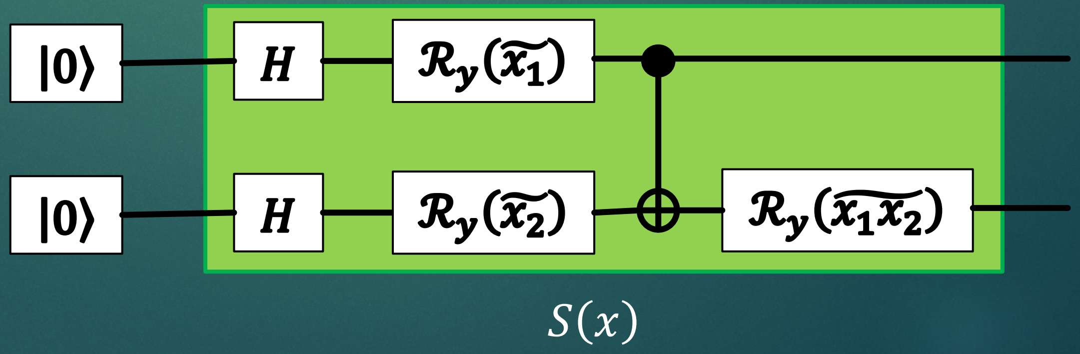
State Preparation

- Basis / Amplitude / Angle encoding
- IQP
- Higher order encoding
- …

Data encoding: example

- $x^i = (x_1^i, x_2^i)$ represents a student:
  - x_1 - university access grade ∈ $[10, 20]$
  - x_2 - hours of independent work per week ∈ $[0, 20]$
- Angle encoding. Normalize to $[-\pi, \pi]$:
  - $\widetilde{x_1} = \frac{x_1 - 10}{10} * 2\pi - \pi \quad ; \widetilde{x_2} = \frac{x_2}{20} * 2\pi - \pi$
- Higher order encoding. Normalize $\widetilde{x_1} * \widetilde{x_2}$ to $[-\pi, \pi]$:
  - $\widetilde{x_1 x_2} = \frac{\widetilde{x_1} * \widetilde{x_2}}{\pi}$





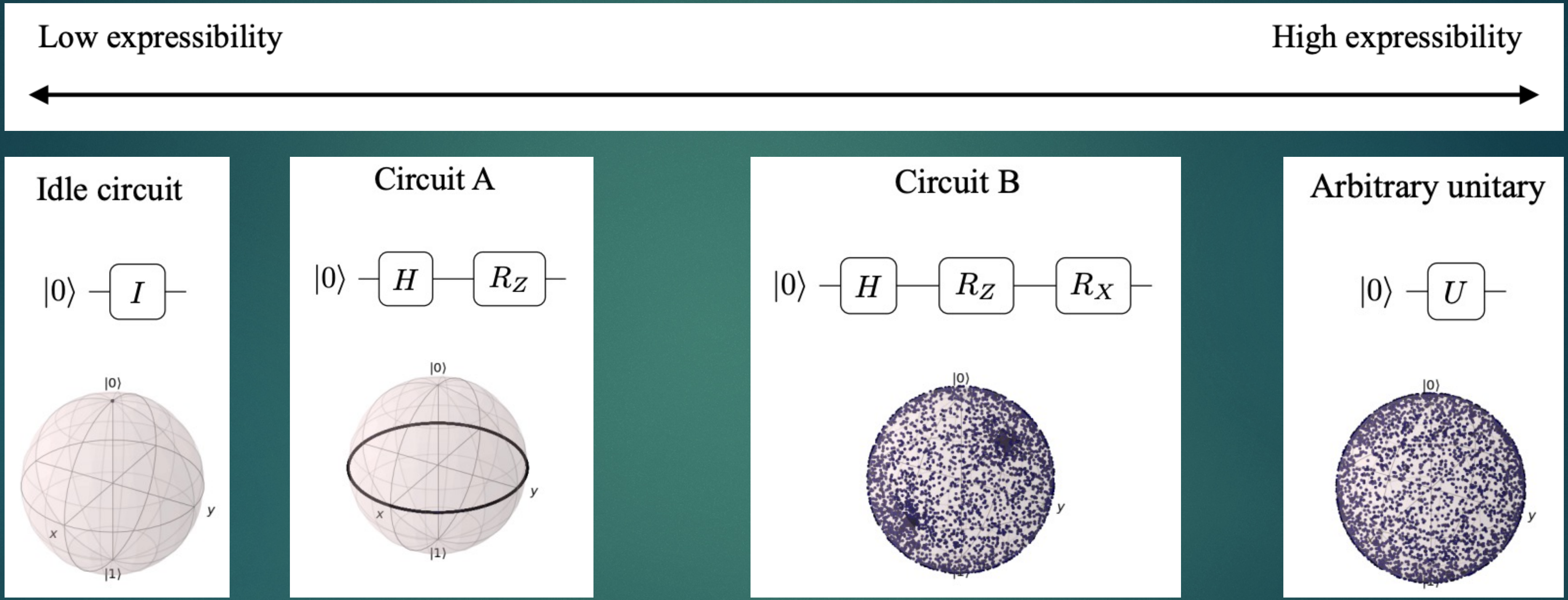
Parameterized model

- What should be the parameterized model (ansatz) ?

$$|0\rangle \rightarrow \boxed{\mathbf{S}(\boldsymbol{x})} \rightarrow \boxed{\mathbf{W}(\boldsymbol{\theta})} \rightarrow \boxed{\boldsymbol{M}}$$

- Problem dependent …
- Open research question …

Parameterized model: expressibility

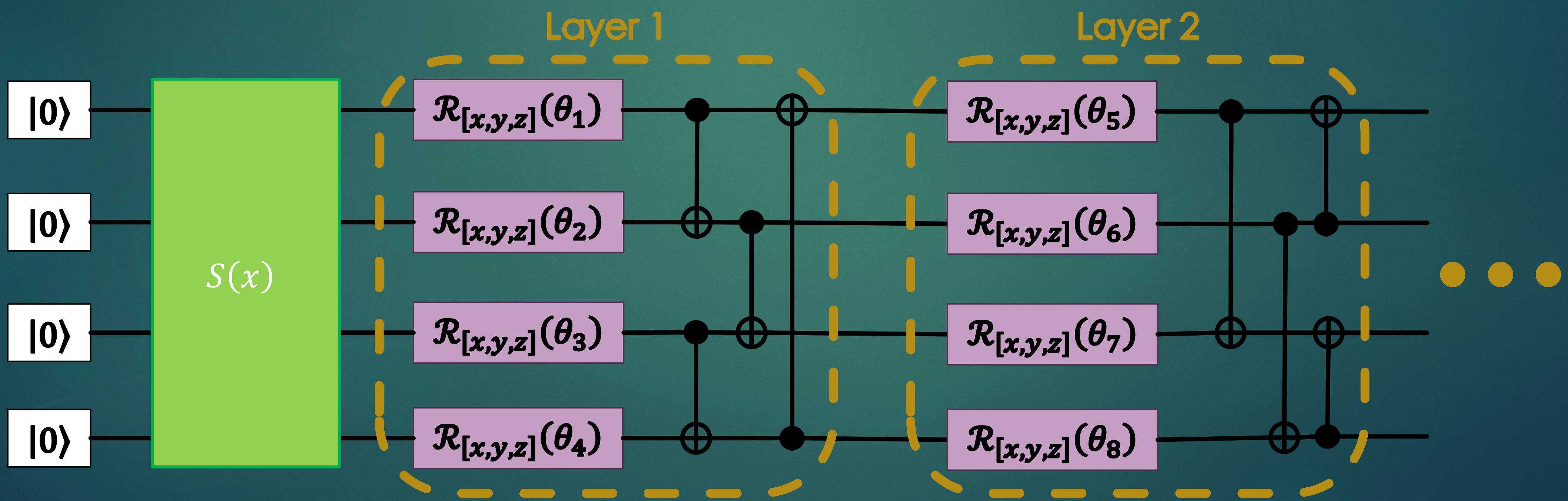


- Expressibility depends on the data encoding and the ansatz
- High expressibility might not always be desirable

[Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms
Sukin Sim, Peter D. Johnson, Alan Aspuru-Guzik; Adv. Quantum Technol. 2 (2019) 1900070]

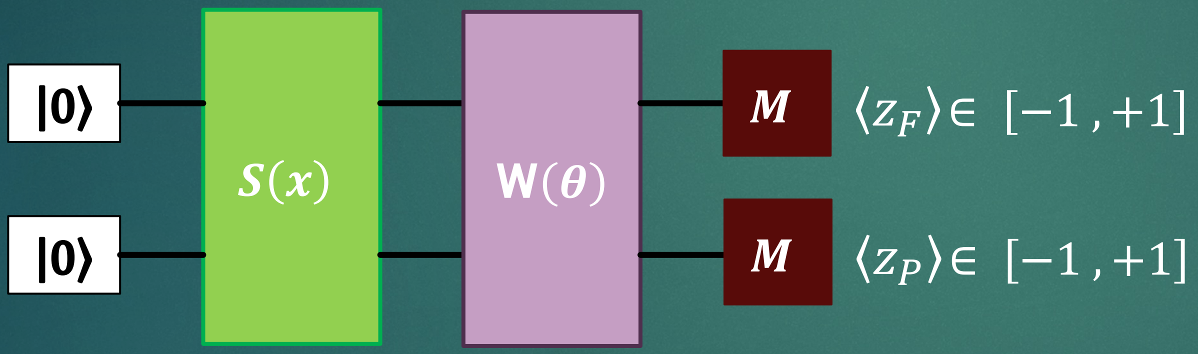
Parameterized model

- Problem dependent versus problem independent
- Hardware dependent versus hardware independent
- Hardware and problem independent common template:



Extracting labels: binary example

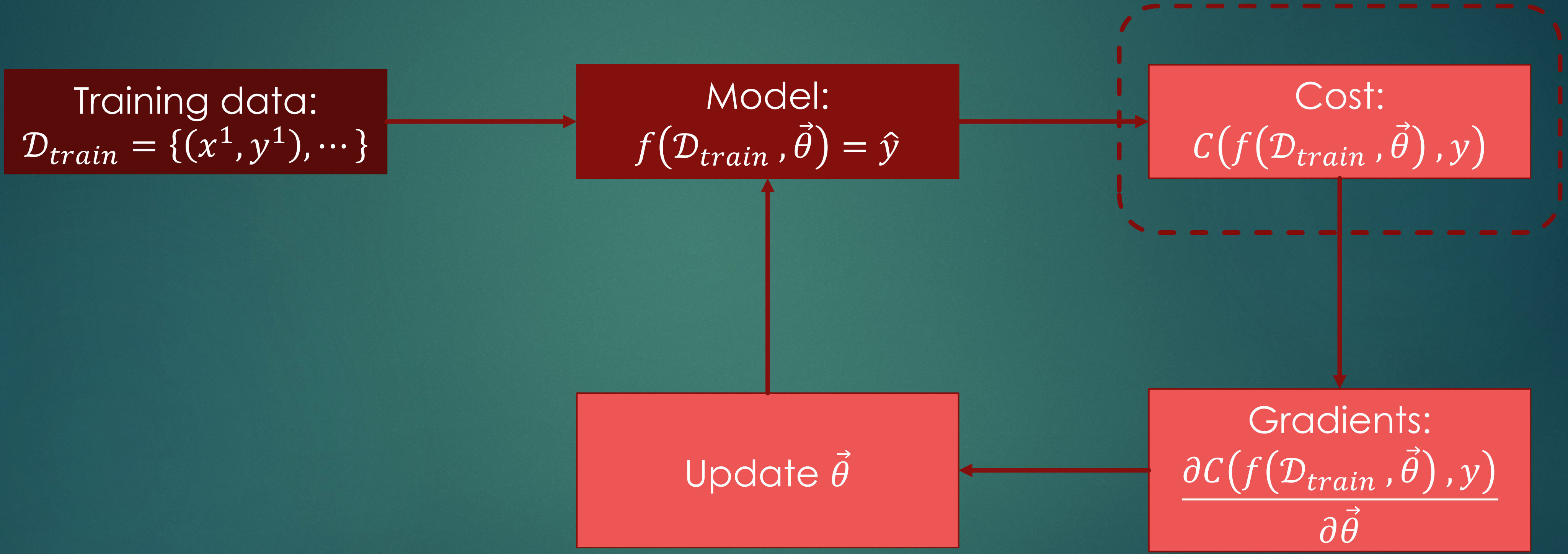
- Binary classification requires distinguishing between 2 labels: y_F and y_P
- On our students example (slide 8): y_F= FAIL ; y_P= PASS
- Two qubit measurement example:



$$\hat{y}_F = \frac{e^{\langle z_F \rangle}}{e^{\langle z_F \rangle} + e^{\langle z_P \rangle}}$$

$$\hat{y}_P = \frac{e^{\langle z_P \rangle}}{e^{\langle z_F \rangle} + e^{\langle z_P \rangle}}$$

Cost Function



Loss function: cross entropy

- The loss function : measurement of the dissimilarity between the true class y^i and the estimated distribution over the C classes $\hat{y}^i$, for data point x^i
- Cross entropy is based on the notion of entropy and compares two probability distributions
  - let y^i be a one-hot encoded vector
    (all 0s, except in the position corresponding to x^i true class):

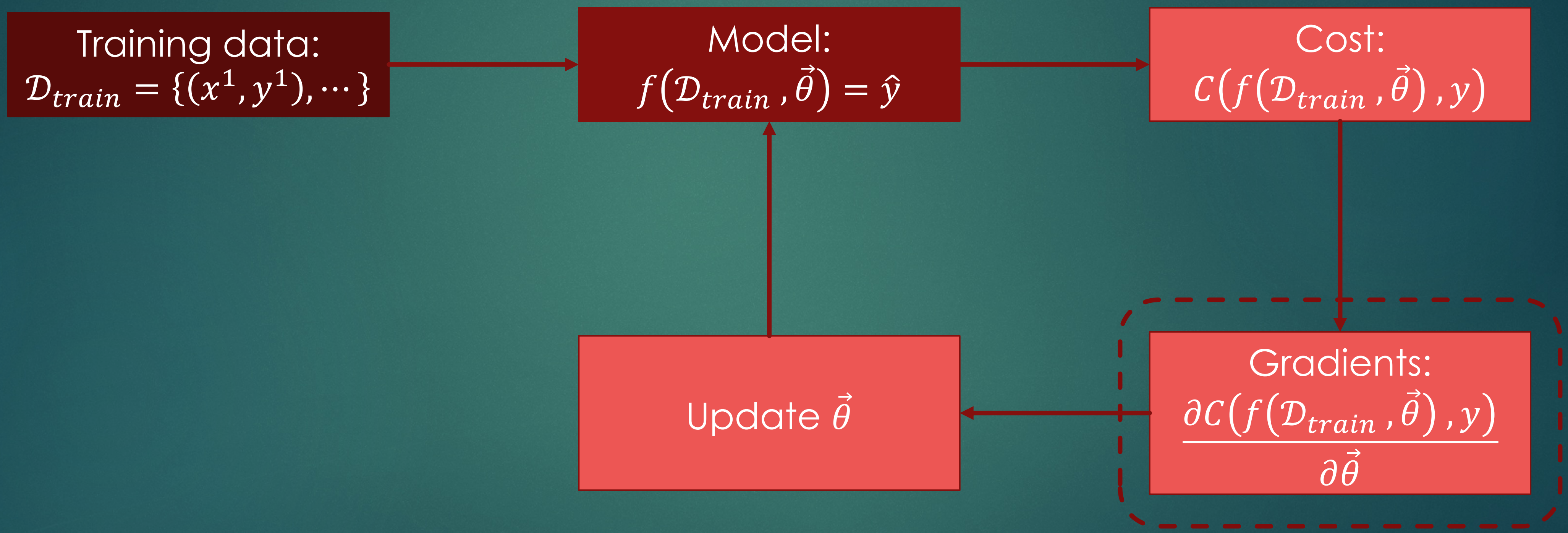
$$ce(y^i, \hat{y}^i) = -\sum_{c=1}^{C} \left(y_c^i * \ln(\hat{y}_c^i)\right)$$

Cost function

- The cost is the average of cross entropy across all M data points:

$$\mathrm{C}(\theta^{t}) = \frac{1}{M}\sum_{i=1}^{M} ce\,(y^{i}, \hat{y}^{i}) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{c=1}^{C} y_c^i * \ln\left(\hat{y}_c^i\right)$$

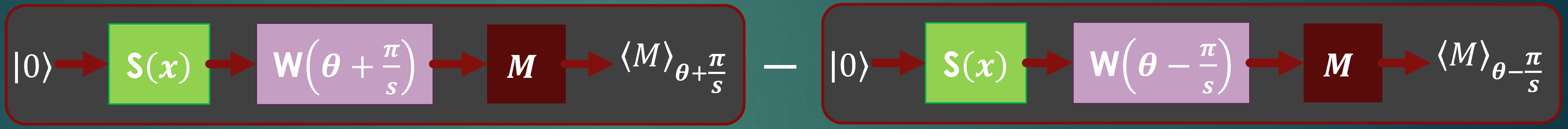
Optimization



Optimization: parameters shift rule

$$\theta^{t+1} = \theta^{t} - \eta\,\vec{\nabla}_{\theta} C(\theta^{t})$$

$\vec{\nabla}_{\theta}\langle O\rangle_{\theta} =$



$$\vec{\nabla}_{\theta}\langle O\rangle_{\theta} = \frac{1}{s}\left[\langle O\rangle_{\theta+\frac{\pi}{s}} - \langle O\rangle_{\theta-\frac{\pi}{s}}\right]$$

For single qubit Pauli gates $s = 2$

$$\vec{\nabla}_{\theta}\langle O\rangle_{\theta} = \frac{1}{2}\left[\langle O\rangle_{\theta+\frac{\pi}{2}} - \langle O\rangle_{\theta-\frac{\pi}{2}}\right]$$

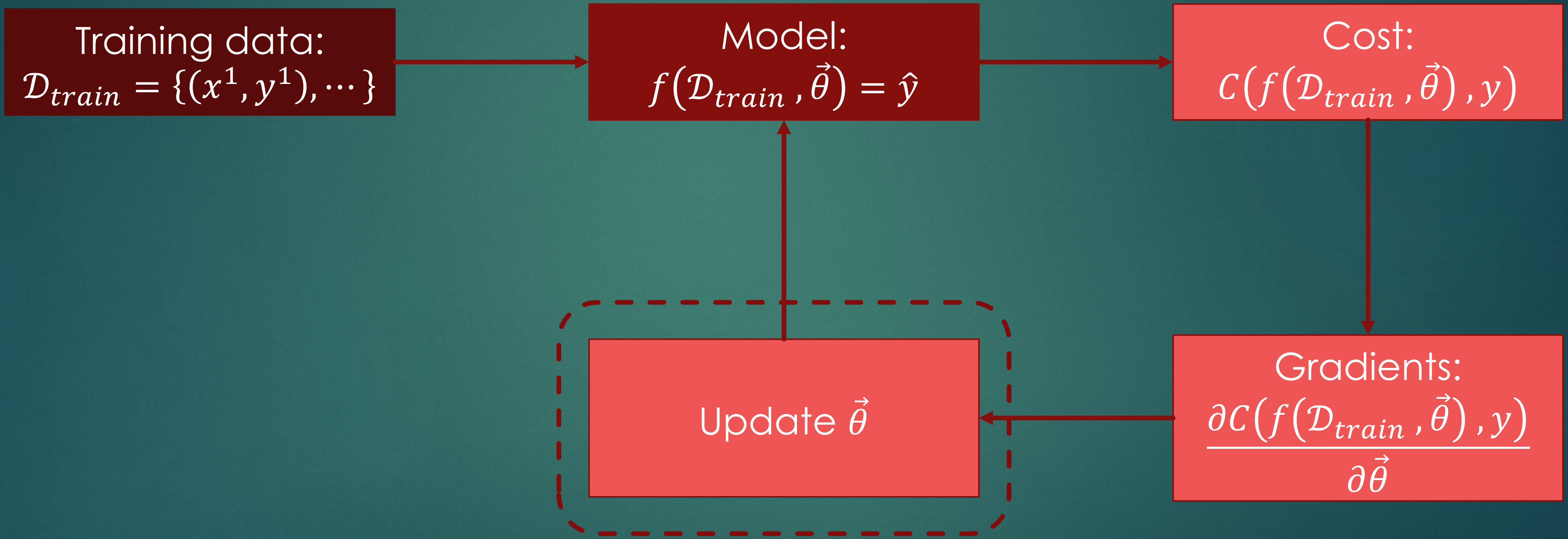
Optimization: parameters shift rule

- The gradient has to be evaluated:
  1. for each parameter θ_k in $\vec{\theta}$
  2. for each data point x^m in the data set
  3. if there are different measurements per class, for each class c

$$\nabla_{\theta_k} \langle \sigma_z \rangle^m_{c,\theta^t_{\partial k}} = \frac{1}{2}\left[\langle \sigma_z \rangle^m_{c,\theta_k + \frac{\pi}{2}} - \langle \sigma_z \rangle^m_{c,\theta_k - \frac{\pi}{2}} \right]$$

- The evaluation of each $\langle \sigma_z \rangle$ requires multiple shots

Iterate


Training data:
$\mathcal{D}_{train} = \{(x^1, y^1), \cdots\}$
Model:
$f(\mathcal{D}_{train}, \vec{\theta}) = \hat{y}$
Cost:
$C(f(\mathcal{D}_{train}, \vec{\theta}), y)$
Gradients:
$\frac{\partial C(f(\mathcal{D}_{train}, \vec{\theta}), y)}{\partial \vec{\theta}}$
Update $\vec{\theta}$

Update θ

$$\vec{\theta}^{\,t+1} = \vec{\theta}^{\,t} - \eta\,\vec{\nabla}_{\theta} C\left(\vec{\theta}^{\,t}\right)$$

$$\vec{\theta}^{\,t+1} = \begin{pmatrix} \theta_1^t - \eta\,\nabla_{\theta_1} C\left(\vec{\theta}^{\,t}\right) \\ \vdots \\ \theta_K^t - \eta\,\nabla_{\theta_K} C\left(\vec{\theta}^{\,t}\right) \end{pmatrix}$$

Variational Quantum Algorithms